Progress Tracking with Version Control and generation of work breakdown structure

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
in
Computer Science Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

by

Neelesh Sharma	**19BCE0768**
Karunesh Tripathi	**19BCE0880**
Daksh Paleria	**19BCE0779**

CSE3001: SOFTWARE ENGINEERING

DECLARATION

June, 2021

I hereby declare that the thesis entitled **Progress Tracking with Version Control and generation of work breakdown structure** submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science Engineering to* VIT is a record of bonafide work carried out by me under the supervision of **Professor Swathi JN** .

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

.

Place : Vellore
Date : 01.06.2021

Neelesh Sharma
Karunesh Tripathi
Daksh Paleria
Signature of the Candidate

ACKNOWLEDGEMENTS

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much Pleasure. We would like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment. In addition, a thank you to Professor Swathi J N, who introduced us to the Methodology of work, and whose passion for the “underlying structures” had lasting effect.Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

DAKSH PALERIA
NEELESH SHARMA
KARUNESH TRIPATHI

Executive Summary

The project is an open source for startups. It is a web-based model for project management. The project is equipped with a login system for users which are being invited to that specific project by the leader as team members who can view the project details, its progress. Therefore, this will help the project admin/leader in maintaining a proper track regarding the project completion. With the help of Version Control, the admin and the fellow team members will be able to refer the commits made by others and point out the mistakes if they found any. The WBS will help the admin in dividing the work and also help them to keep the track of the work as that would help them in maintaining the work load and do get rid of merge conflicts if found.

Table of Contents

List of figures

List of table

List of Abbreviations

- SRS - Software Requirements Specification
- WBS -Work Breakdown Structure
- VSC -Version Control
- PT - Progress Tracking

1.Introduction

1.1Objective

The objective of the Software Design Specification is to describe the specific design of the Project management system with WBS and Version control system. The design specification includes an overview of the design along with software module decomposition. This document provides a detailed description of each software module's design. For each module, a user interface design and class diagram design is given. As well, a process description is described for each module. It is in the process description that the details of what logic will need to be implemented are given.

1.2Motivation

We ourselves as student developer find it difficult in managing our tasks and update our progress while we work as team, we have to rely on different platforms and update our progress accordingly on those platforms.

1.3Background

Our tool will consist of a version control which will help the users to commit/change/update any data without any hassle, we have took some inspiration from Git which is also a version control for GitHub, since Git is open source we were able to see how git functions smoothly and helps to reduce work, a version control also helps the administrator/leader to see the commits/changes in the tasks and it even helps the leader to see who made the most recent commit and the complete history of that specific or any commit.

The tool will also consist of an WBS, Progress tracking functionality which will help the admin, all fellow members in keeping track of all the work. We aim to make our project completely open source in order to make things much better by accepting ideas/contribution from fellow developers.

2.Project Description and Goals:

It is within the scope of the Software Design Specification to describe the specific system design of the Project management system with WBS and Version control project. This would include user interface design, object-oriented class design, process design, and data design. Any specific detail that is needed about the standards or technology used to design the software are within the scope of this document. It is outside the scope of this document to describe Project management system with WBS and Version control systems and technology. It is also outside the scope of this document to describe in any detail at all how certain mentioned standards or technologies work and operate. The tool will consist of WBS, Progress tracking, Version Control (VCS) functionality which will help the admin, all fellow members in keeping track of all the work. We aim to make our project completely open source in order to make things much better by accepting ideas/contribution from fellow developers.

3.Functional Requirements

3.1Login/Registering

3.1.1Introduction

The Project management software with WBS and VSC shall allow a user to register/login themselves via a mail, password and a unique project ID given to them by the project admin. The user then accordingly inputs his/her email and password to gain access over the project on platform.

3.1.2Functional Requirements

Purpose*:* Sending the user authentication message about whether they logged in or not
Input*:* Take email, password and project ID from the user.
Processing*:* Verify the email is correct and then verify whether the password matches with the given email or not. Then check whether the project ID exist or not
Output*:* Successful Authentication message

3.1.3Stimulus Response

User actions	System actions
(1) Input mail, password, project ID	
	(2) Check whether the mail, password and project ID exists or not
	(3) If email, password matches and the project ID exist then generate a success message and let the user login
	(4) Let the user Log In
	(5) If email, password doesn’t matches and the project ID exist then generate a error message
	(6) If email doesn’t exist then take the user to register page
(7) Ask for email and password	(8) Validate the email and password and let the user register and navigate them to login page.

3.2Work Break-Down Structure (WBS) and UML

3.2.1Introduction

The software has a feature called WBS which allows the project admins to edit the timeline and setup the milestones for all the given tasks. Only project admin can add a task and change its deadline, whereas the team members and just view the tasks and update the progress for same.

3.2.2Functional Requirements

Purpose*:* Set up deadline for the project and milestone for each task
Input*:* Email, Username, Task name, deadline.
Processing: Verify whether the username exist in the project management software or not and make a chart accordingly.
Output*:* Make a chart for the given deadlines of tasks.

3.2.3Stimulus Response

User actions	User response
(1) Enter task name	
(2) Enter its deadline	
(3) Enter the username	
	(4) Verify if Username exists in project group or not
	(5) Make a chart accordingly for the tasks and let the team mates view the deadline.

3.3Version Control (VSC)

3.3.1Introduction

This feature allows the admin to keep the track of all the necessary changes easily and lets the user push a new file or remove or push some code to an existing file very smoothly. This even allows the admin and the respective team mates to find out who was the last person to commit the code in any given file.

3.3.2Functional Requirements

Purpose*:* Allowing the user to commit the codes to the platform smoothly and securely which further allows the admin to get complete history of the commits.
Input: Email, password, action like add new file, remove file, push code to an existing file, create branches.
Processing*:* Process the user option very carefully in order to avoid any conflicts with the changes made by other users.
Output*:* Success message or error message depending upon the error type that got reflected on when the user performed a specific request in VSC.

3.3.3Stimulus Response

User actions	User response
(1) Call authentication function to validate the mail, password of the user	
	(2) System Checks for presence of email
	(3) System reverts back by giving options to user about what to do (add, remove, push to exist)
(4) User chooses the desire option among the given ones	
	(5) VSC read the option of the users and reverts back to ask the user to input commands into it.
(6) User enters the command necessary to perform the action.	
	(7) The VSC then executes the commands and makes sure that there is no conflict in the codes of all the team mates and reverts back with a success message

4.DESIGN APPROACH AND DETAILS.

4.1List of figures:

Fig 4.1.1 ER diagram

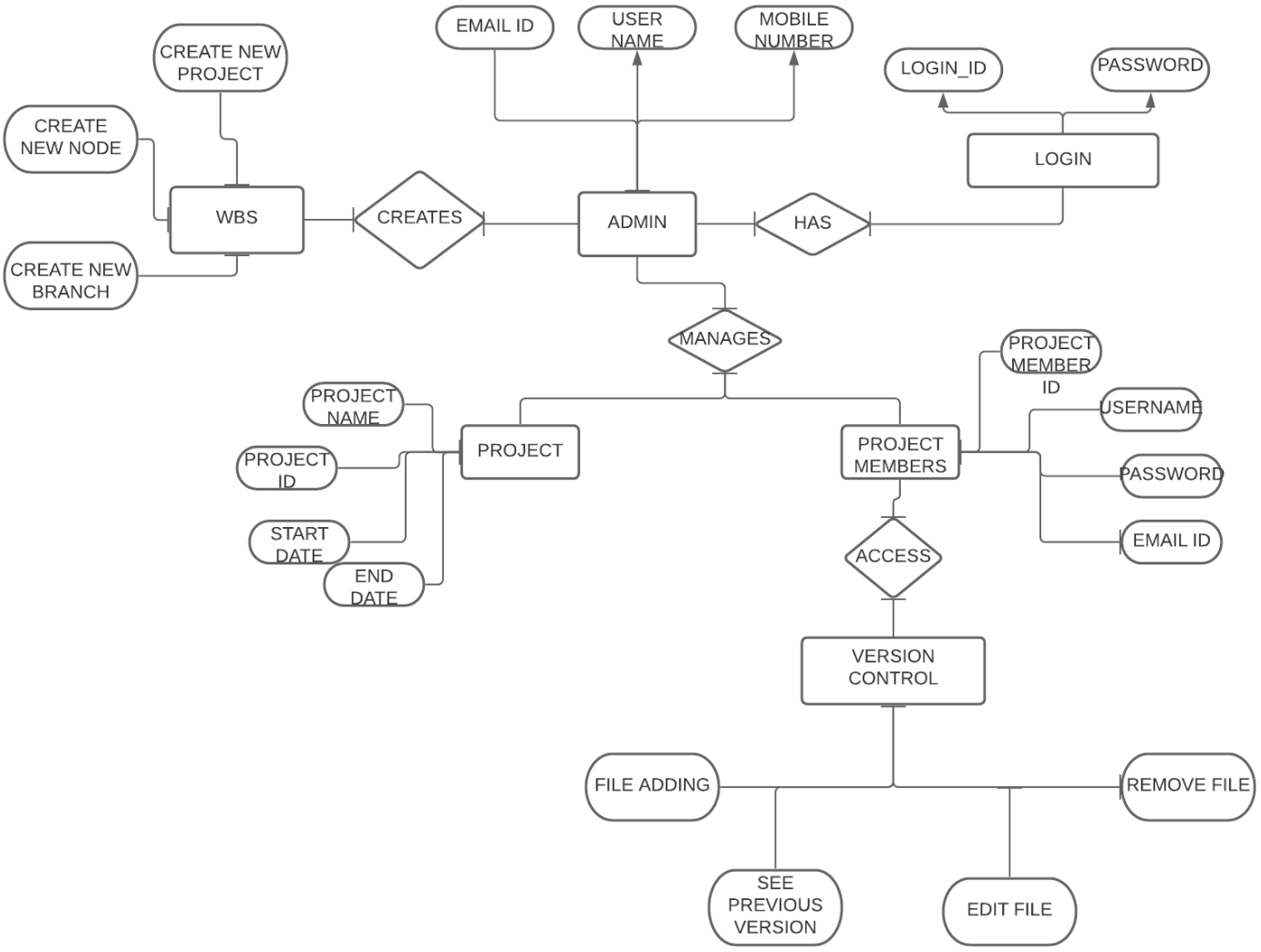


Fig 4.1.2 Data Flows

The following figures represent the data flow diagrams of the Electronic Stamp software. The first data flow diagram, figure 2, is the top level data flow. This is followed by the more detail data flow of the Email Client Software

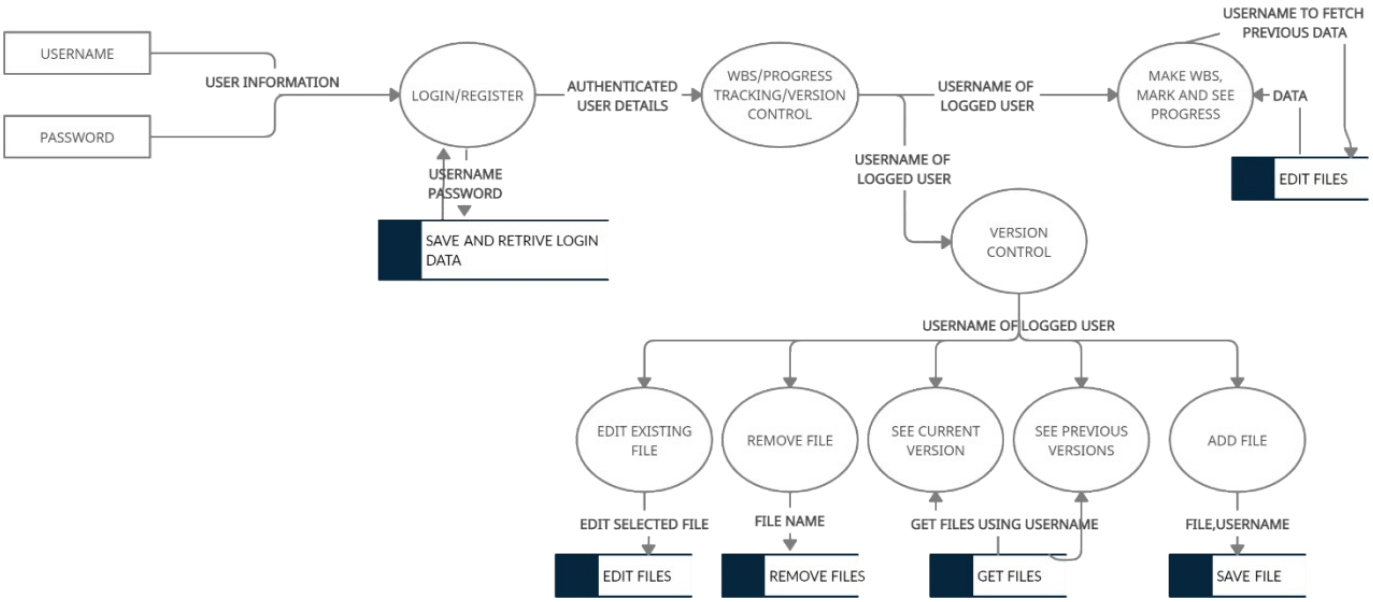


Fig 4.1.3 Use Case Diagram

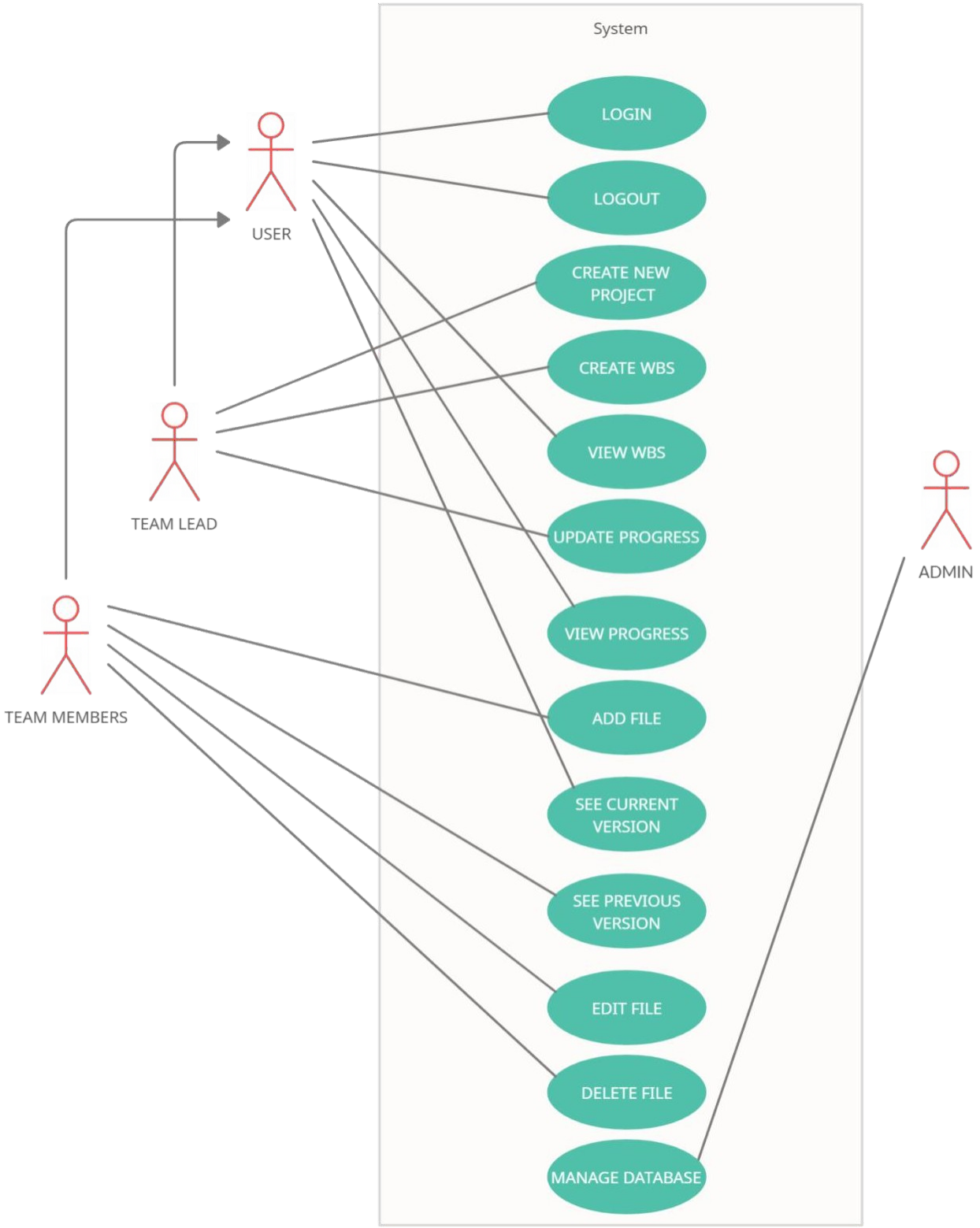


Fig 4.1.4 Swinglane Activity Diagram

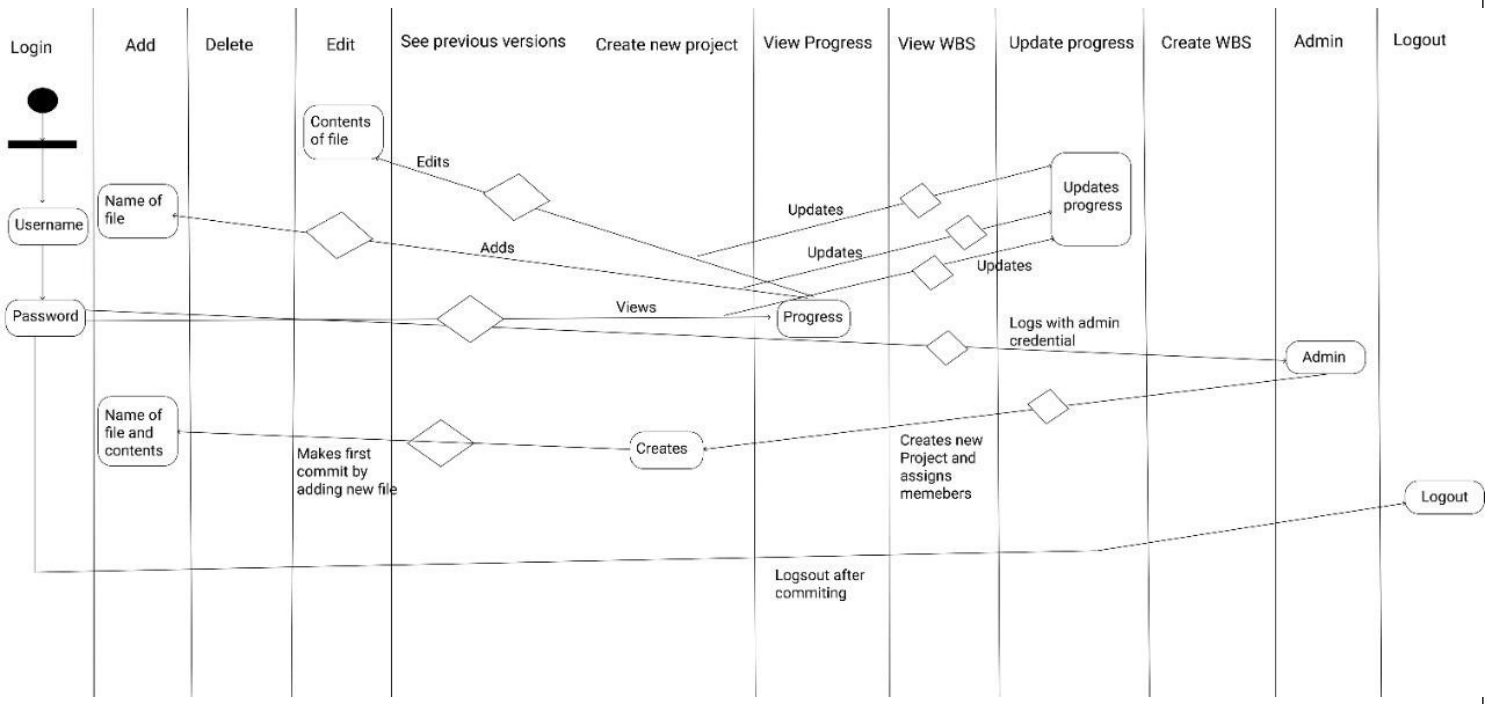


Fig 4.1.5 Login and Authentication

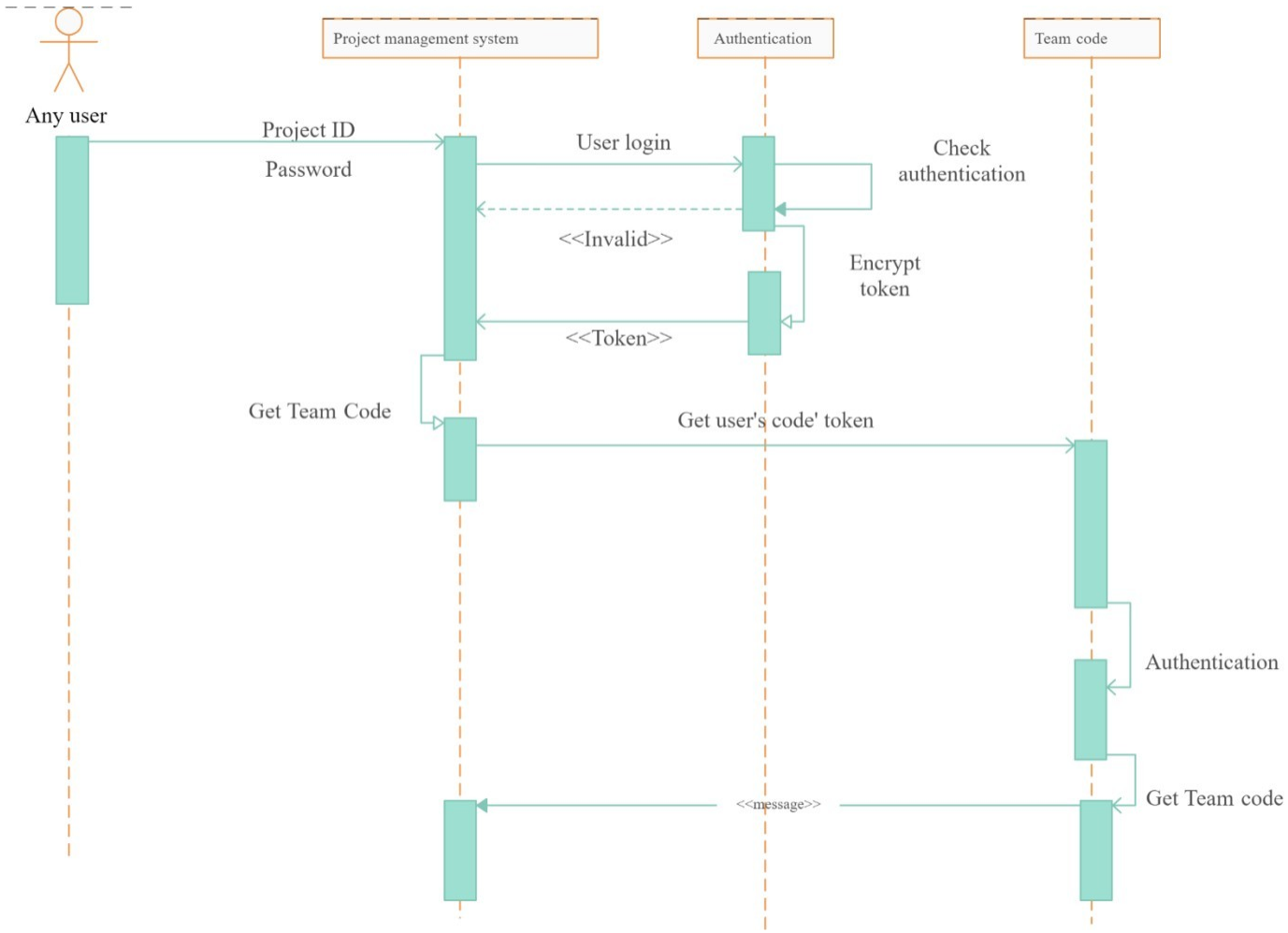


Fig 4.1.6 Progress status generation

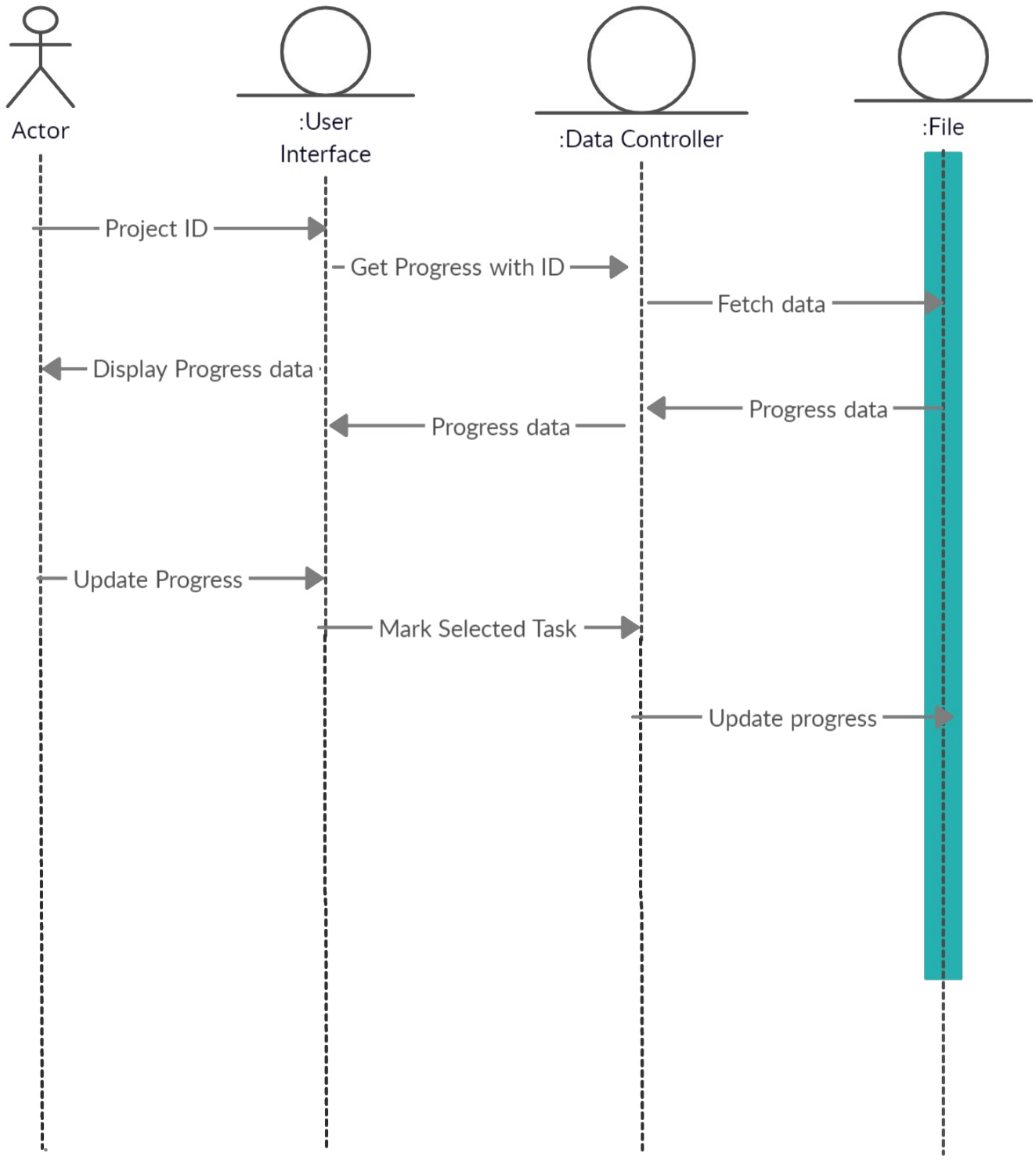


Fig 4.1.7 Generation of WBS

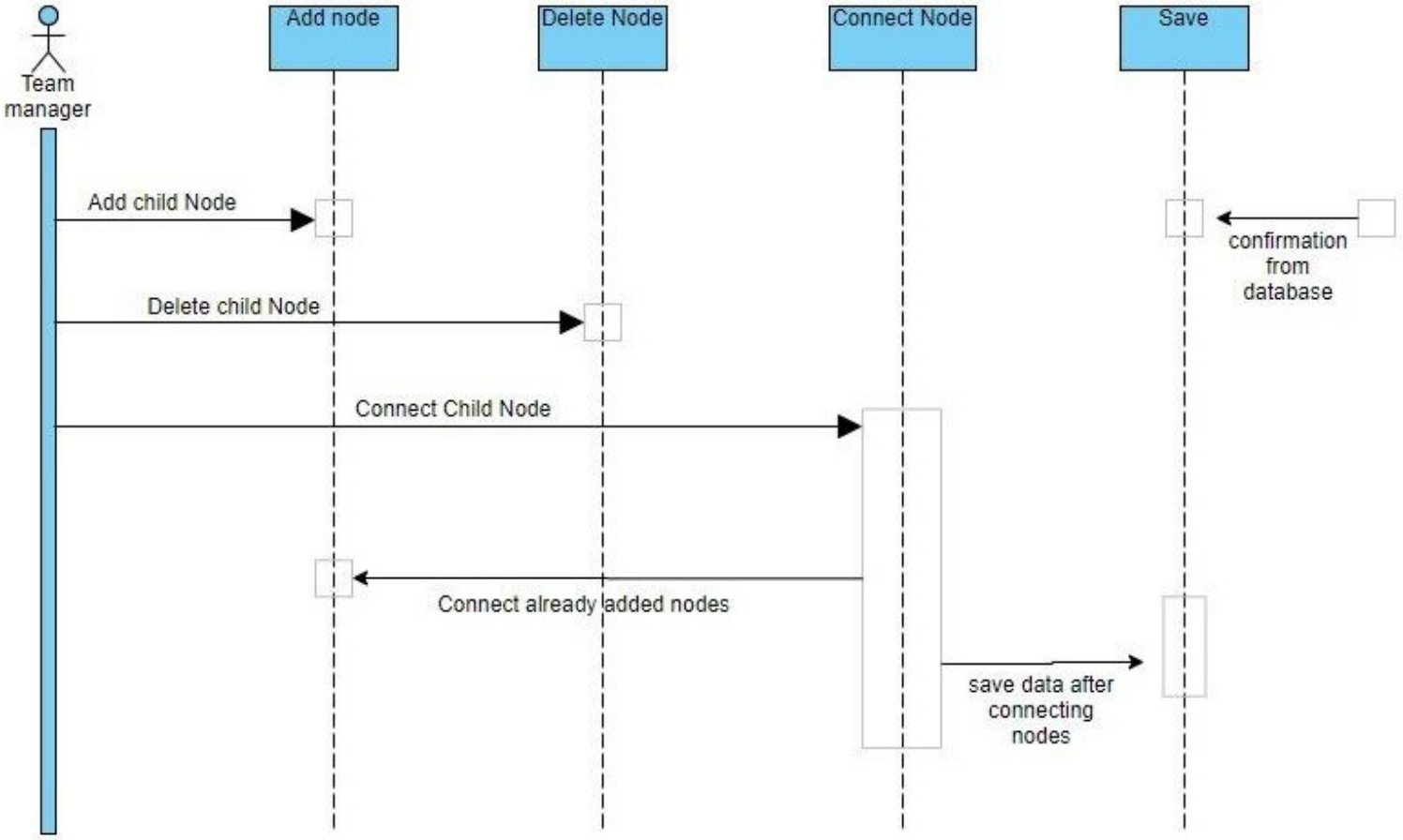


Fig 4.1.8 Create new project

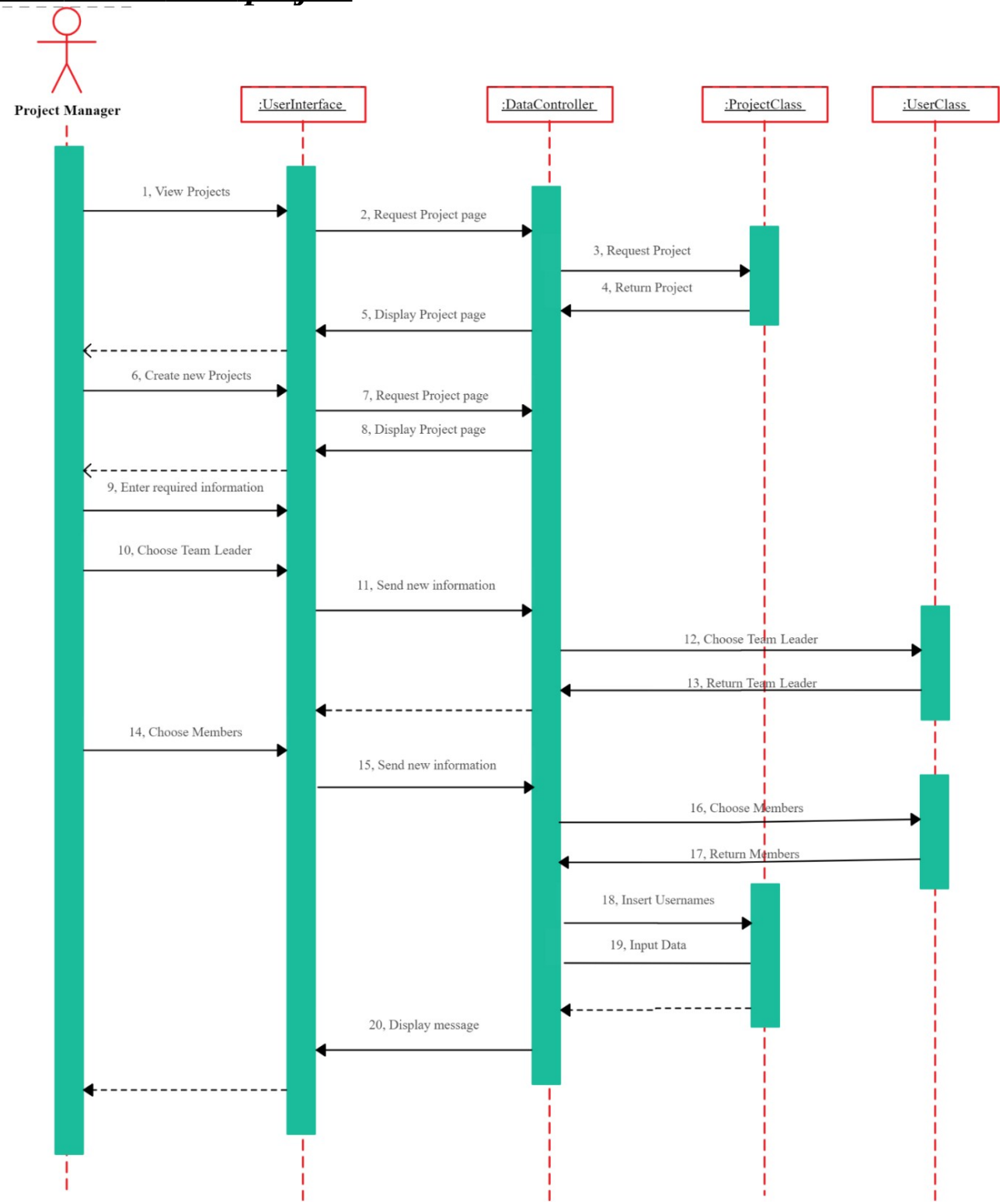


Fig 4.1.9 View, Add., Delete File in Version Control

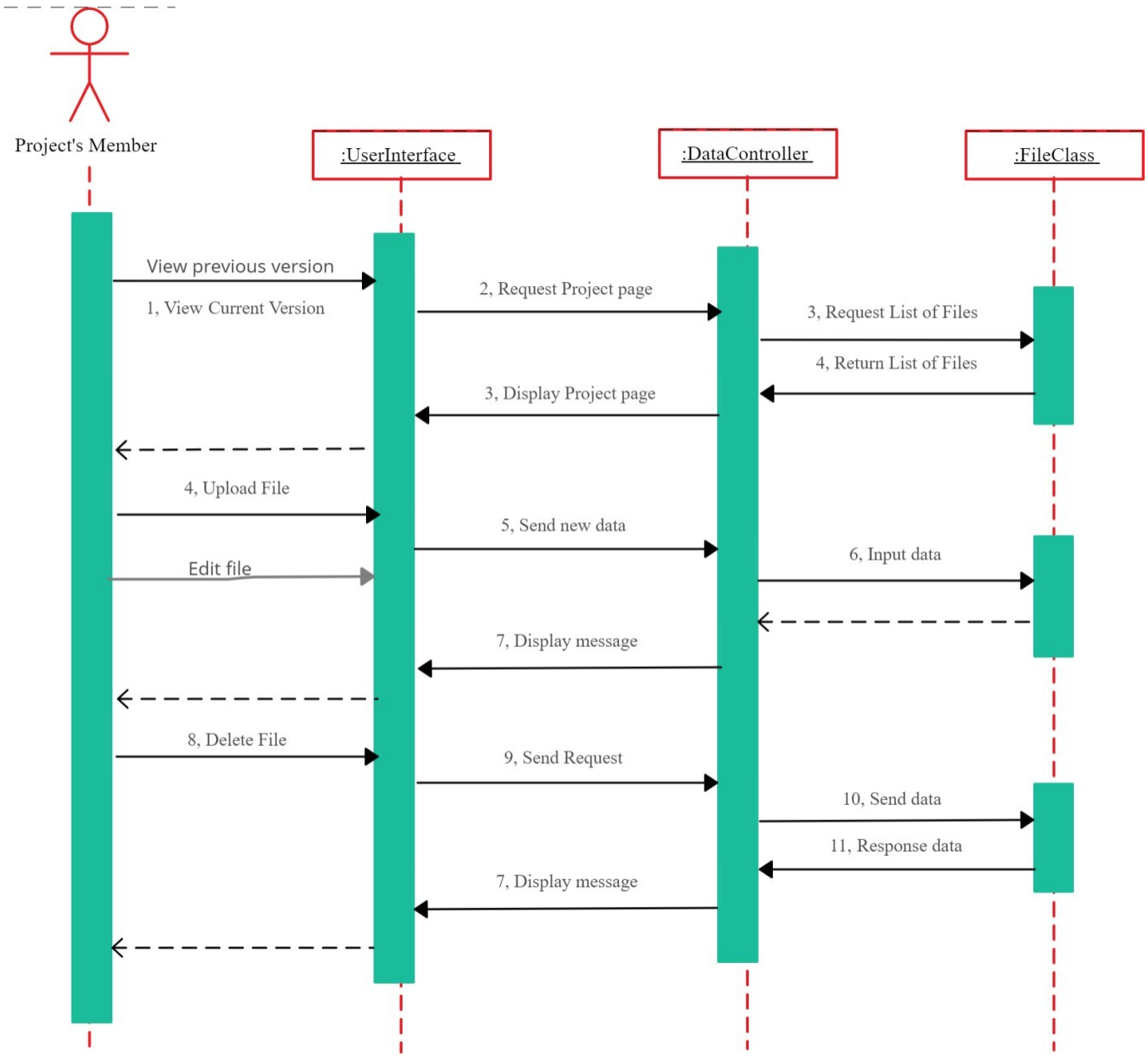


Fig 4.1.10 Login

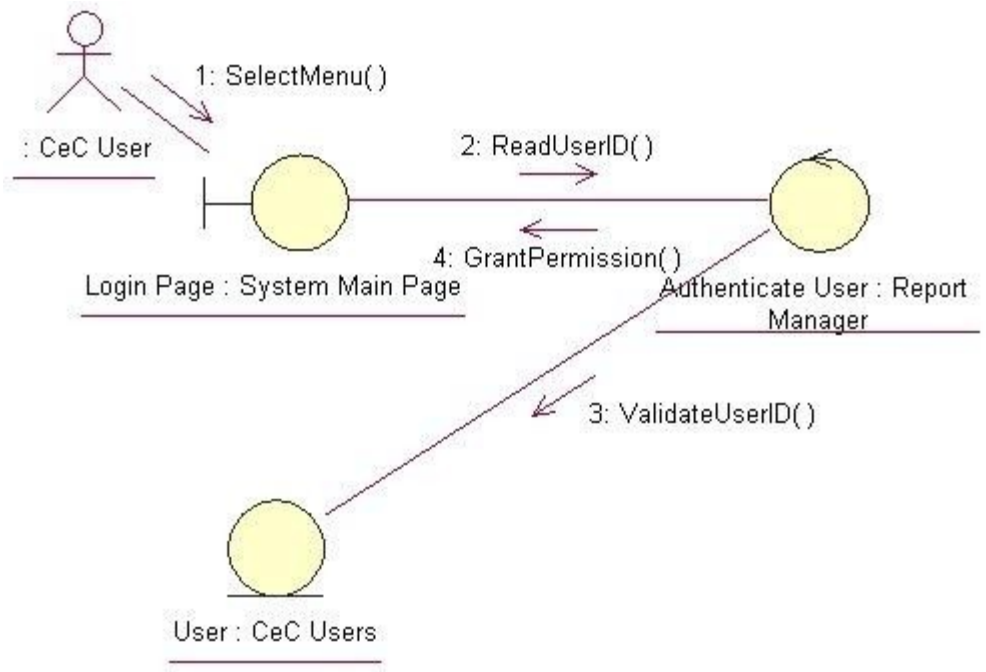


Fig 4.1.11 Logout

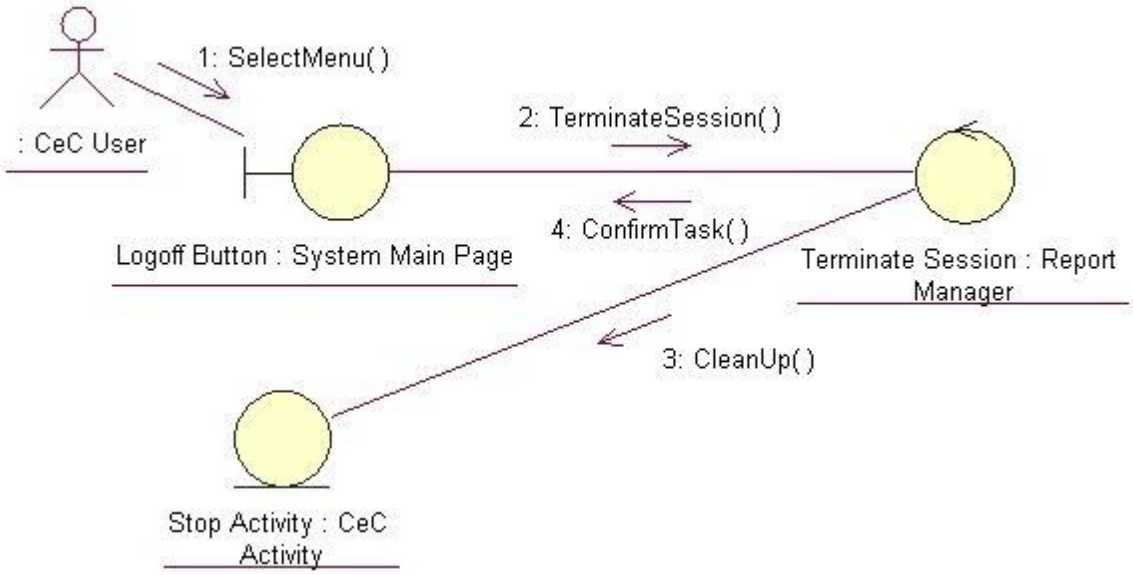


Fig 4.1.12 WBS Generation

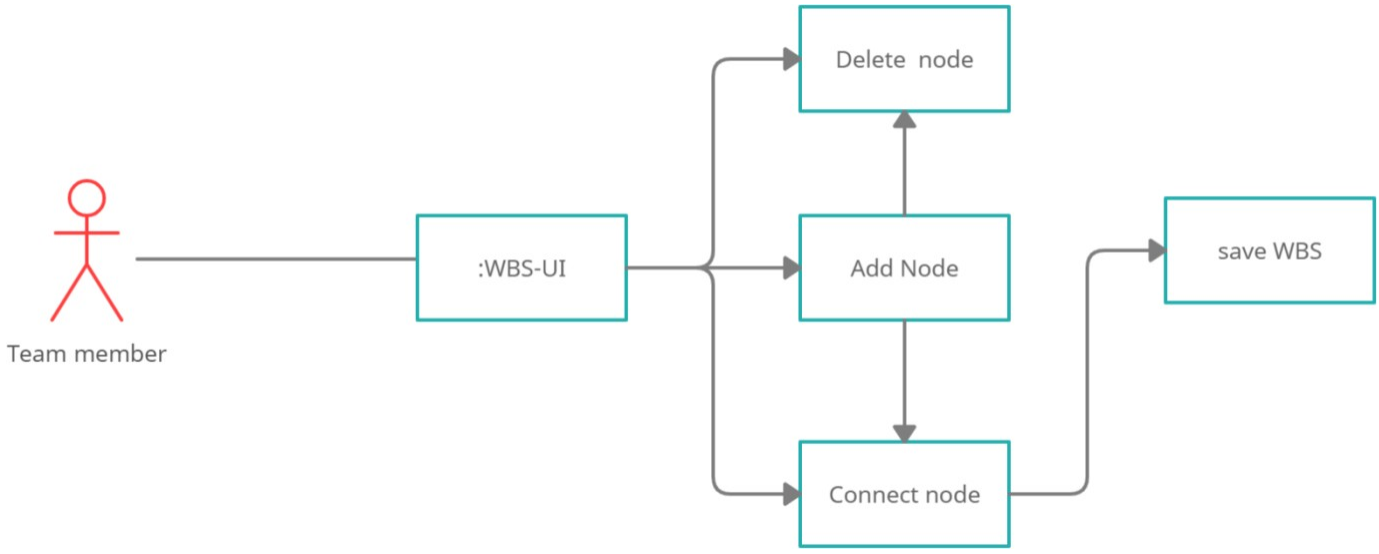


Fig 4.1.13 Progress Tracking

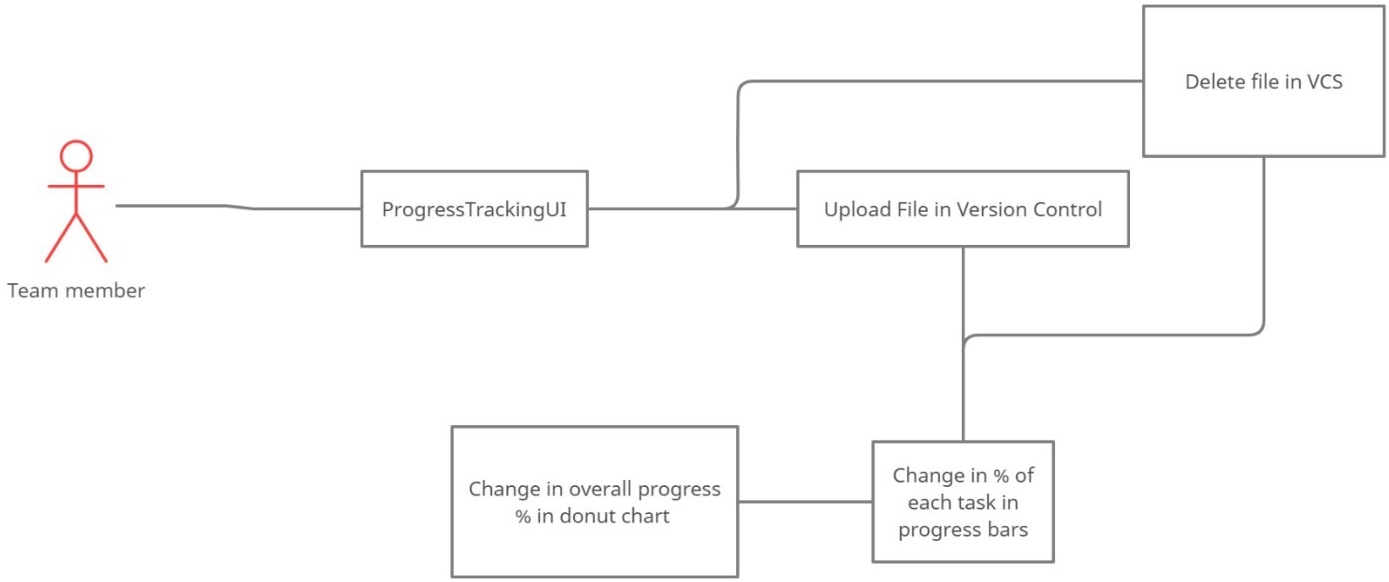


Fig 4.1.14 View File

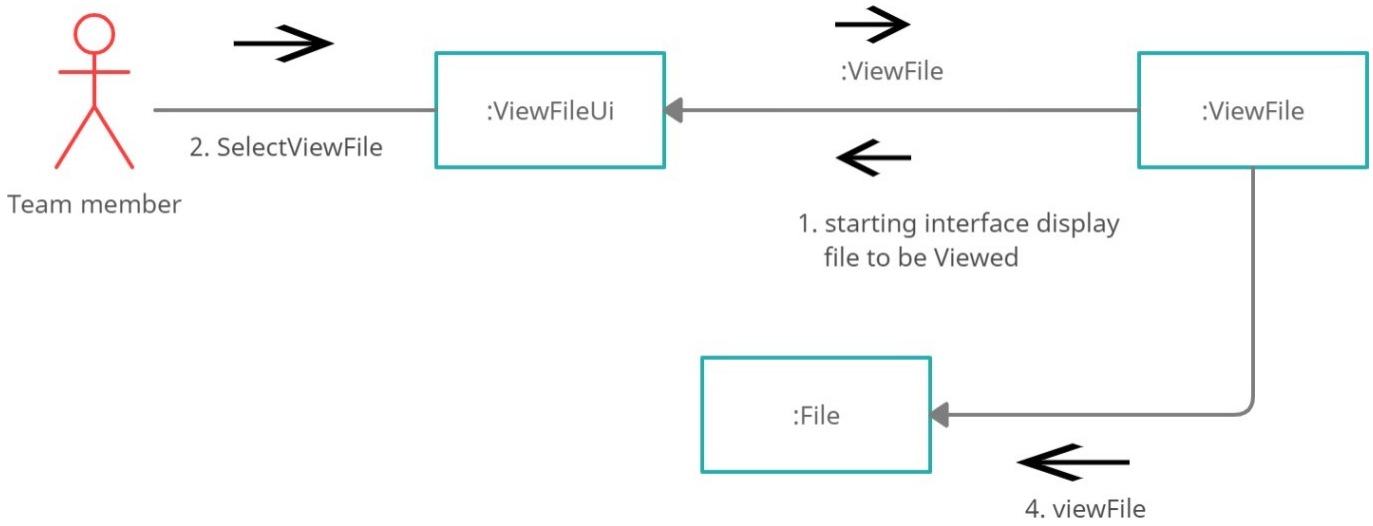


Fig 4.1.15 Add File

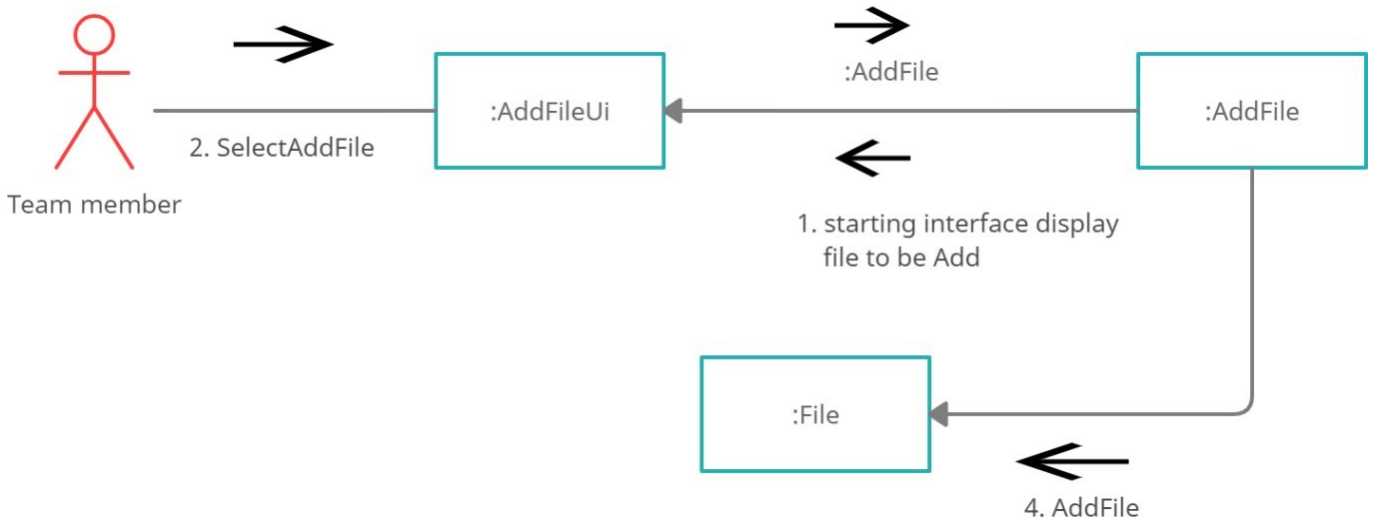


Fig 4.1.16 Edit FIle

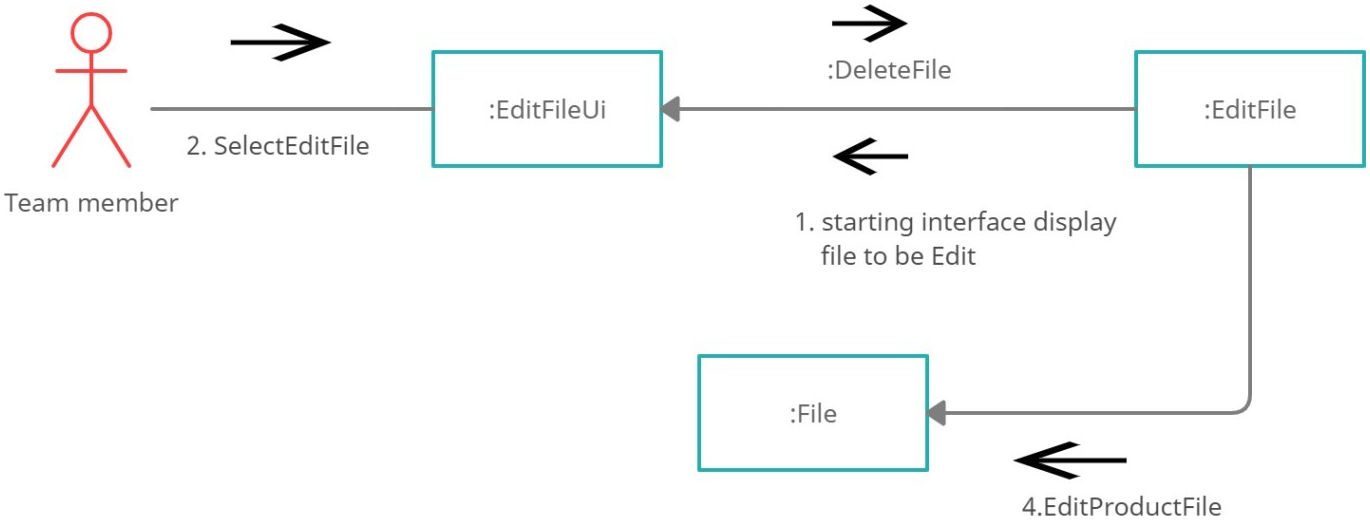


Fig 4.1.17 Delete File

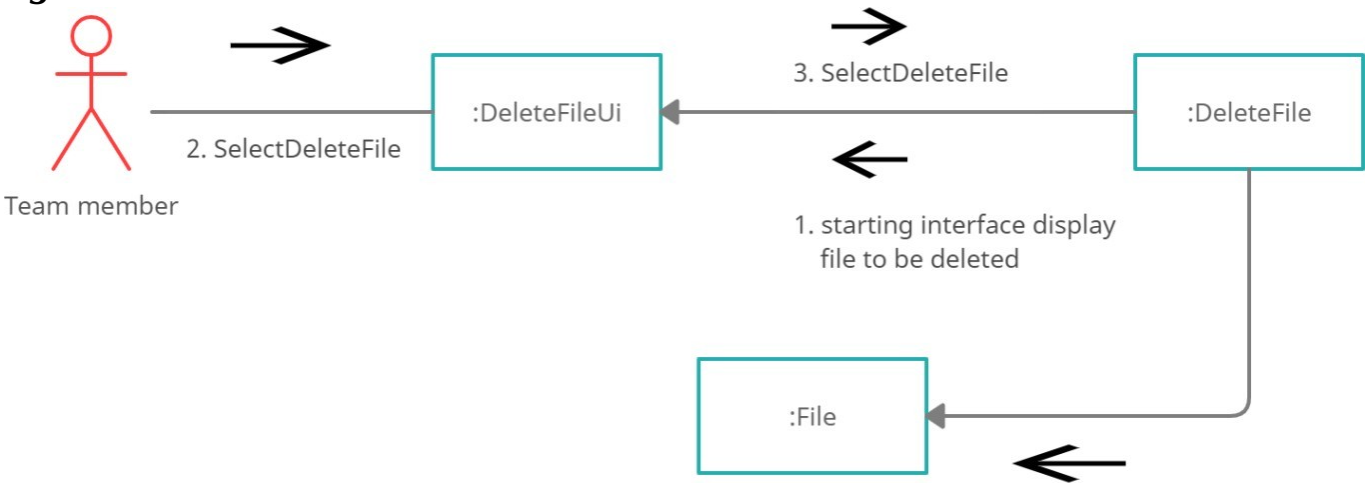


Fig 4.1.18 Class Diagram

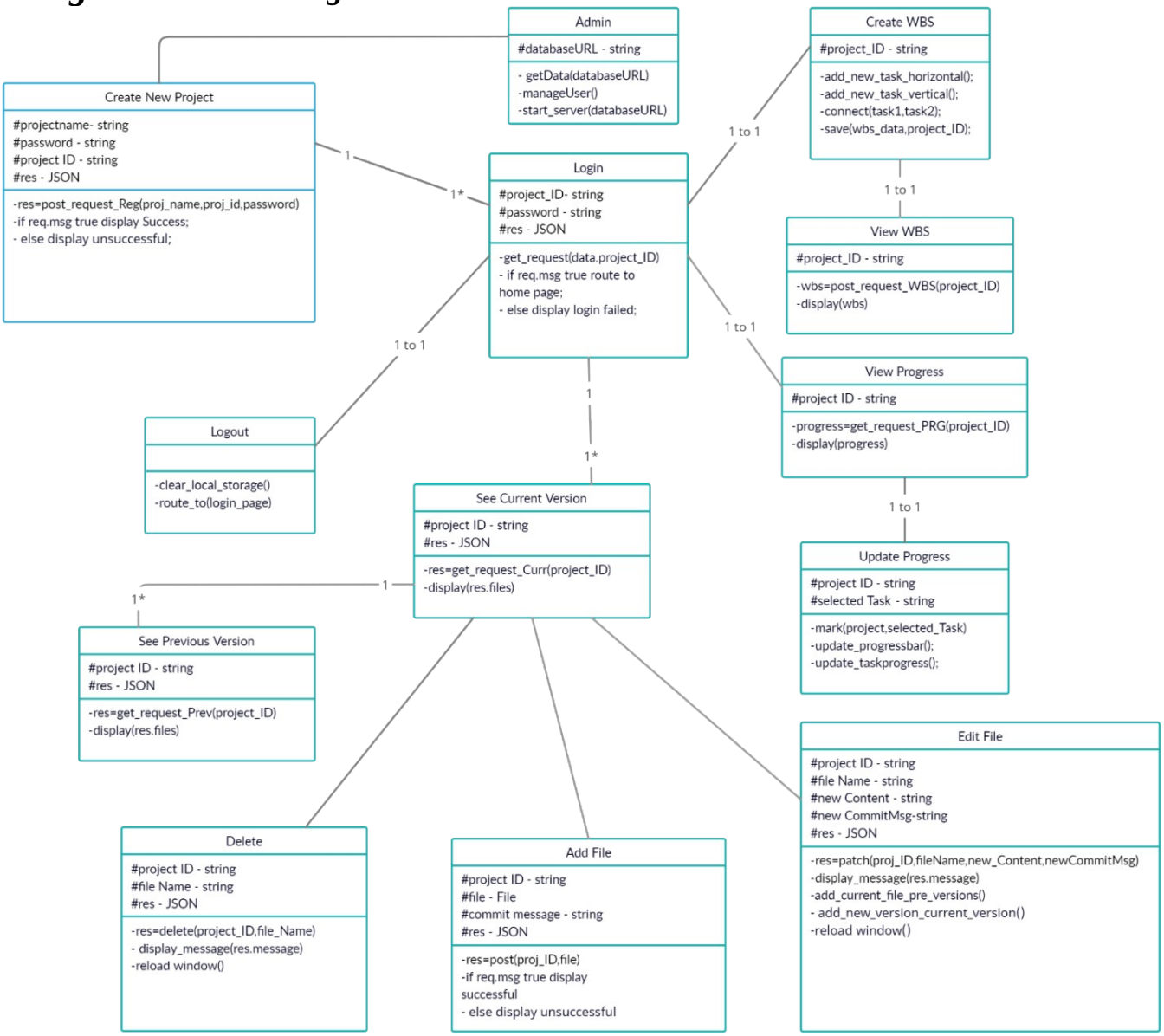


Fig 4.1.19 State Transition Diagram

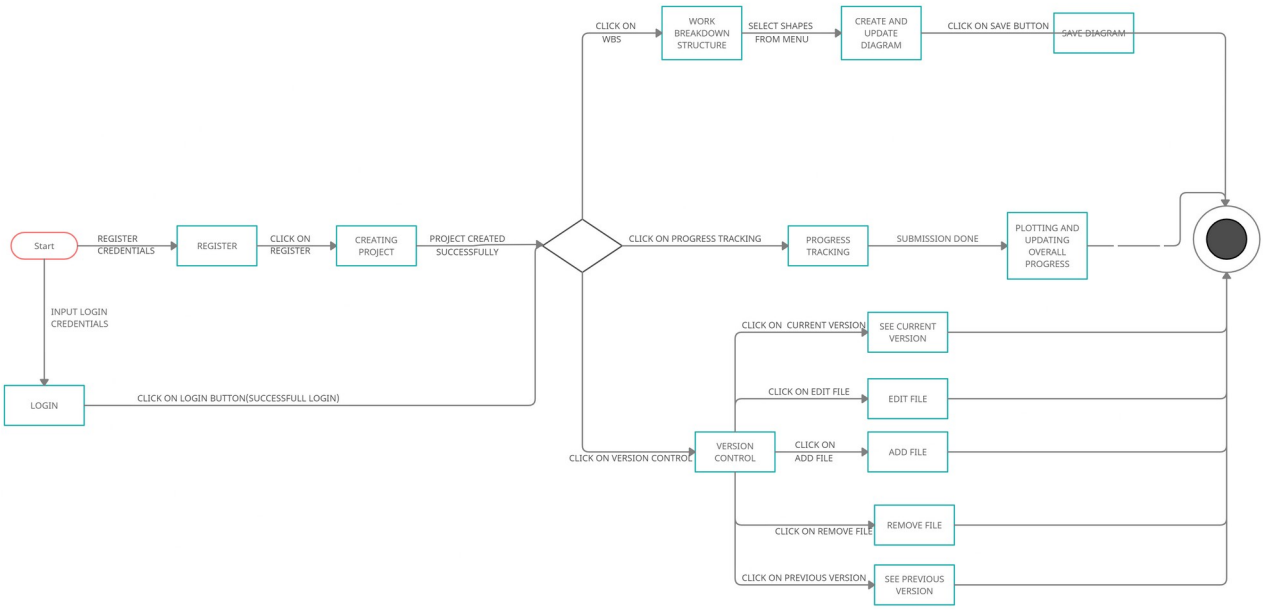
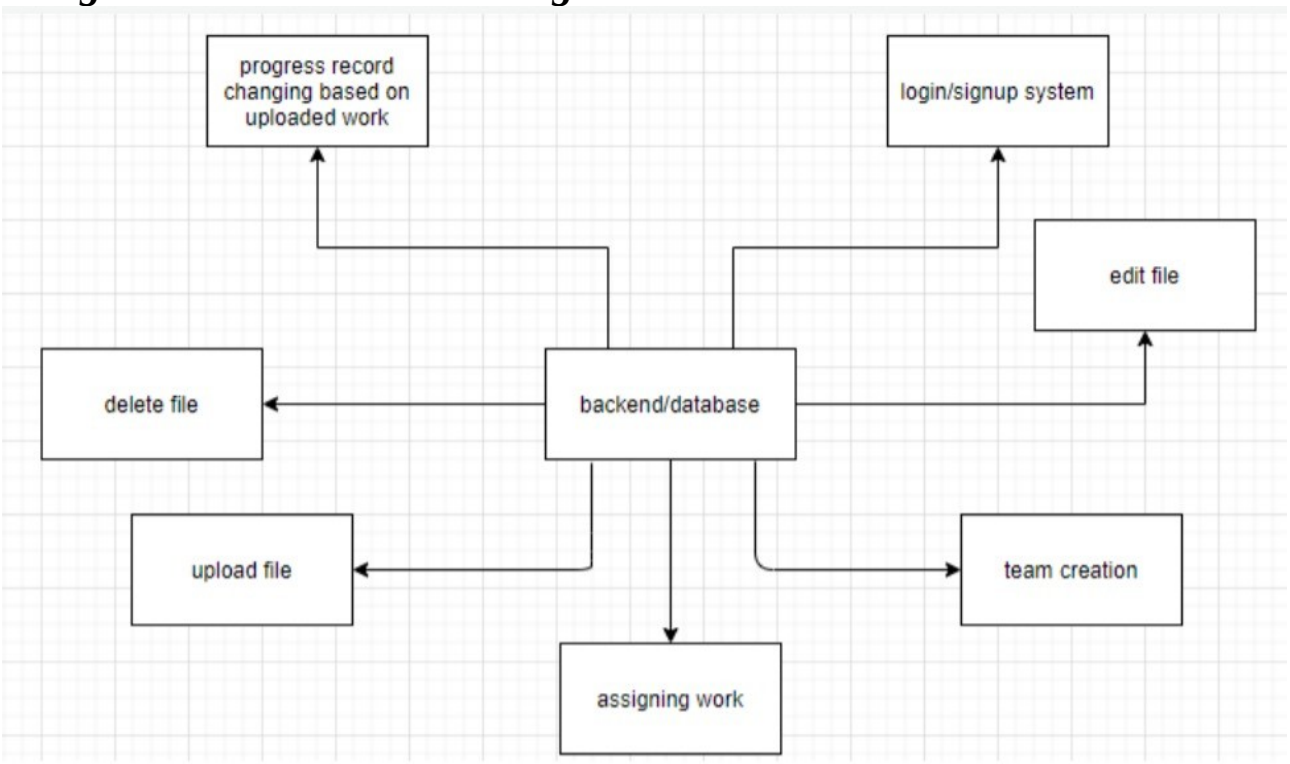


Fig 4.1.20 Architectural Design



REPOSITORY MODEL

4.2Codes and Standards.

The software has been developed using HTML,CSS ,bootstrap and javascript. reactJS, nodejs.

Libraries used :

•React
•Fragment
•BrowserRouter
•Switch
•react-router-dom
•Link
•useState
•useEffect
•useHistory
•PropTypes
•diagram-library
•diagram-library-react
•express
•express-fileupload
•ejs
•fs
•mongoose
•jquery
•util
•console
•http
•path
•body-parser
•cors
•yarn
•npm

API used :

● Self Create REST API using MongoDB server

Software versions used:

•“HTML”: 5
•“CSS”: 2.1
•“ReactJS”:17.0.1
•“NodeJS”:12.0
•"express": 4.17.1
•"jQuery": 3.6.0
•"mongoose": 5.12.1
•"nodemon": 2.0.7
•“MongoDB”:4.4.5
•“Windows”:10

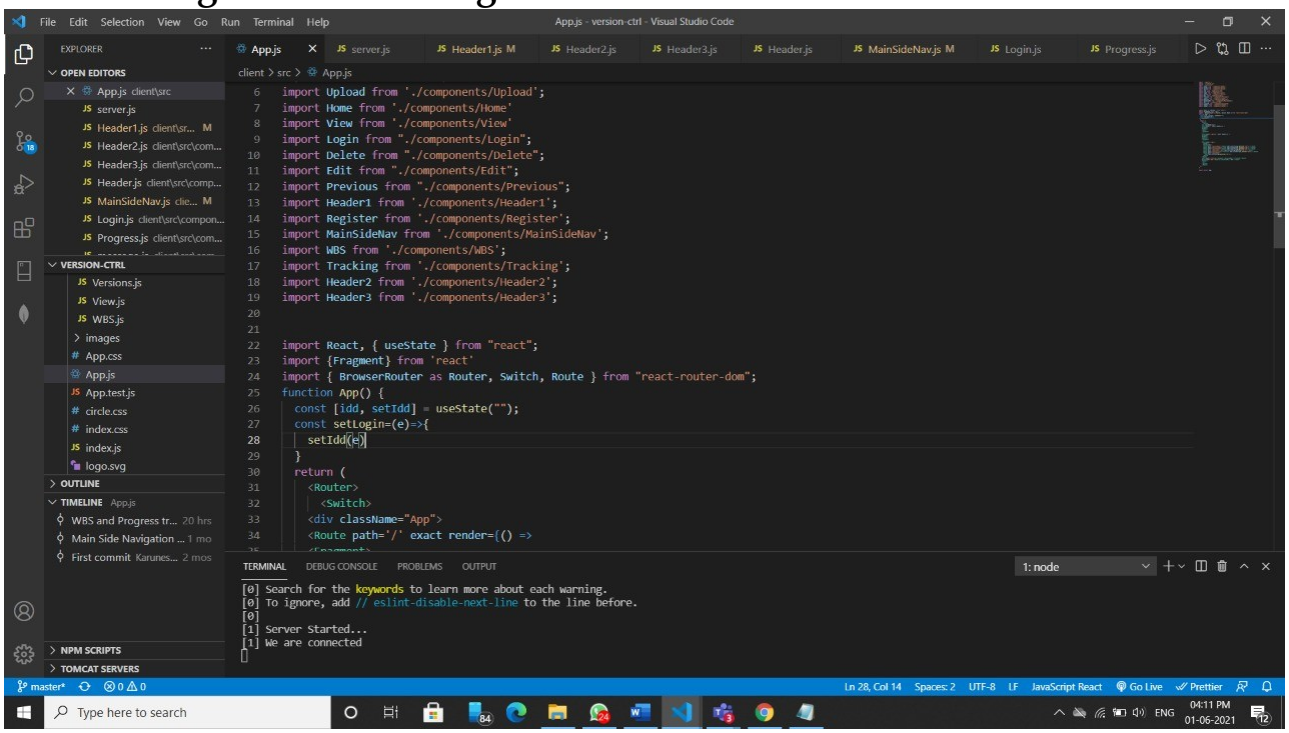
Standards followed:

● Global variables have a limited use
● Standard headers for different modules have been mentioned
● All naming conventions for local, Global variables have been followed
● Functions, dependencies and Libraries have been clearly stated and named.
● Proper Indentation has been followed throughout the codes.
● Exception and error handling measures have been taken for both Backend and Frontend codes.
● No identifier has a multiple usage
● GOTO statements are not used,
● Codes are well documented
● High Cohesion, low coupling used
● Modularity maintained by reusing functions to create multiple cards and info pages.

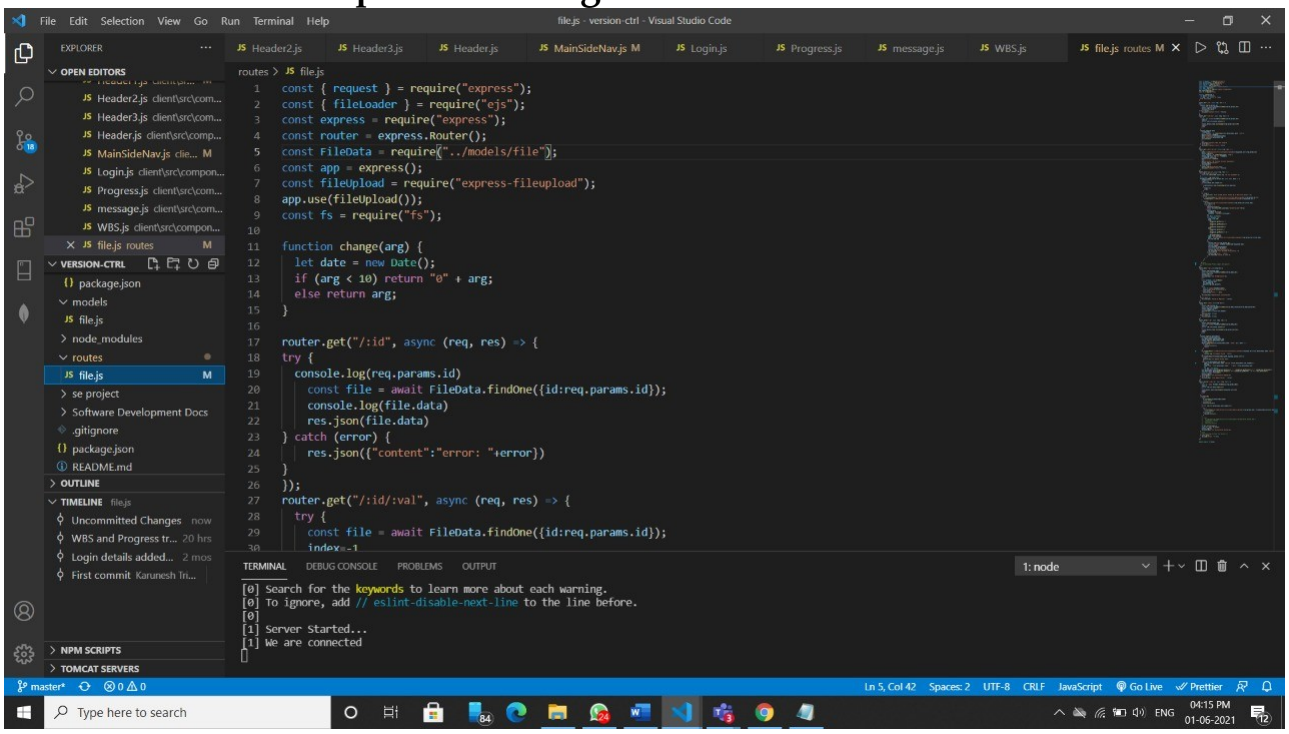
5.CODE SNIPPETS

5.1BACKEND CODE:

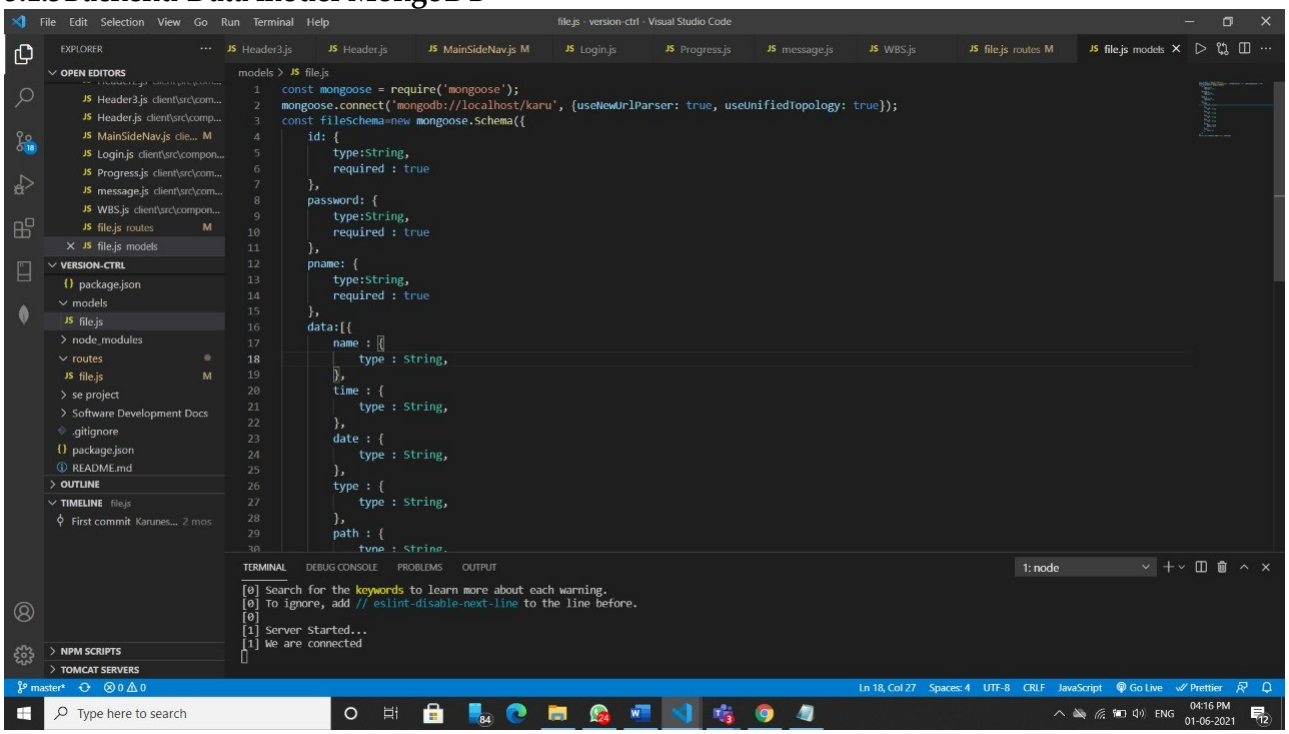
5.1.1Routing and Positioning



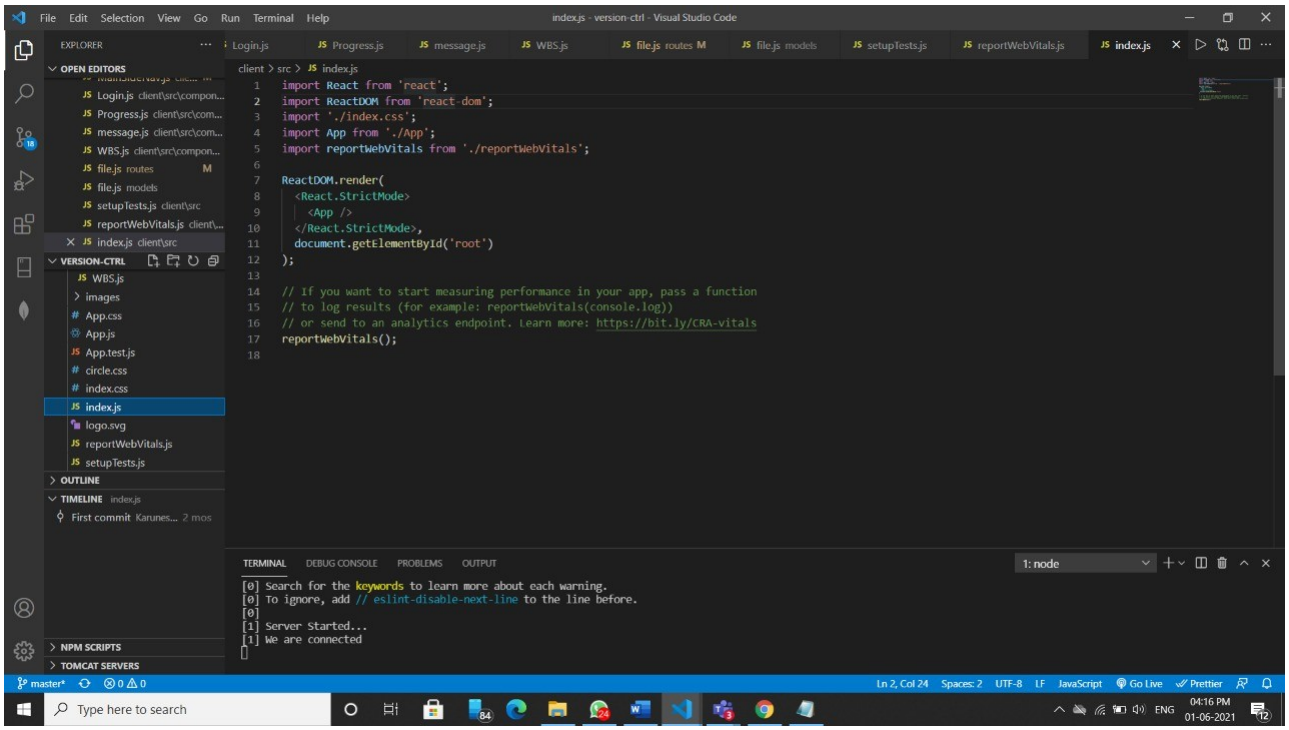
5.1.2Backend API request handling



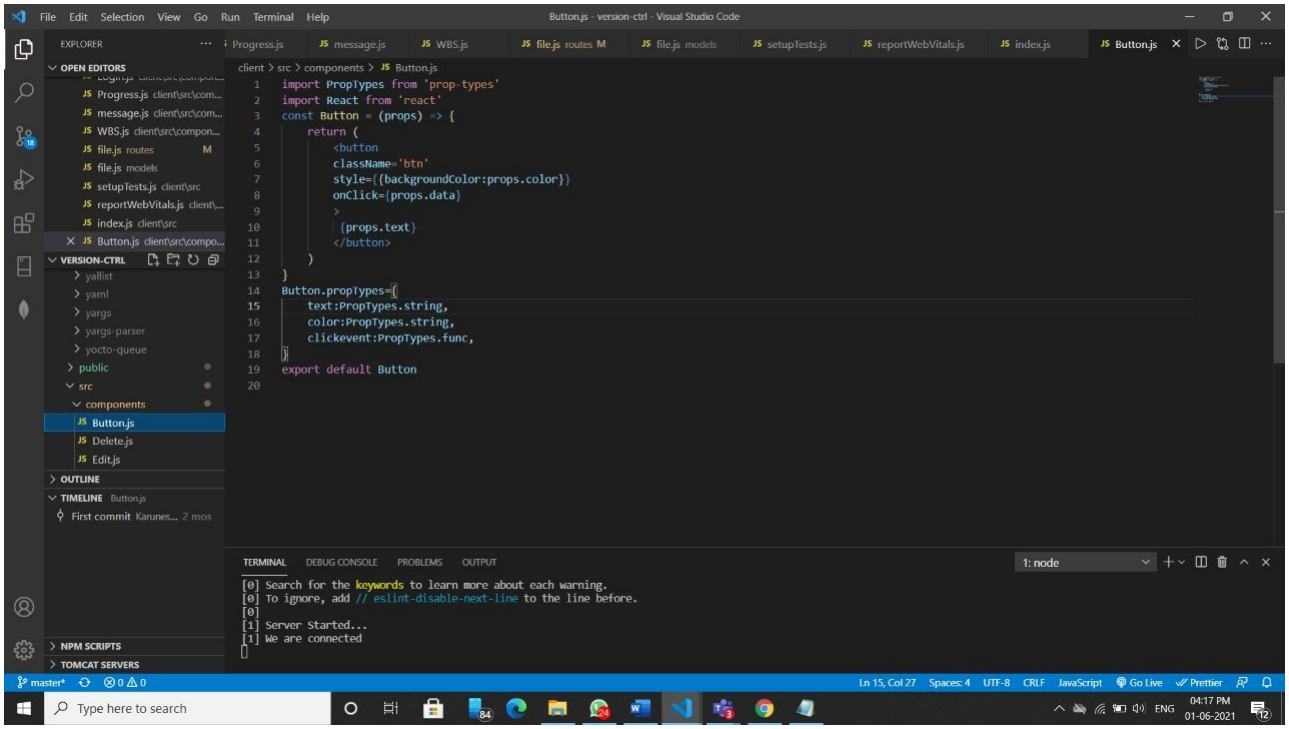
5.1.3Backend Data model MongoDB



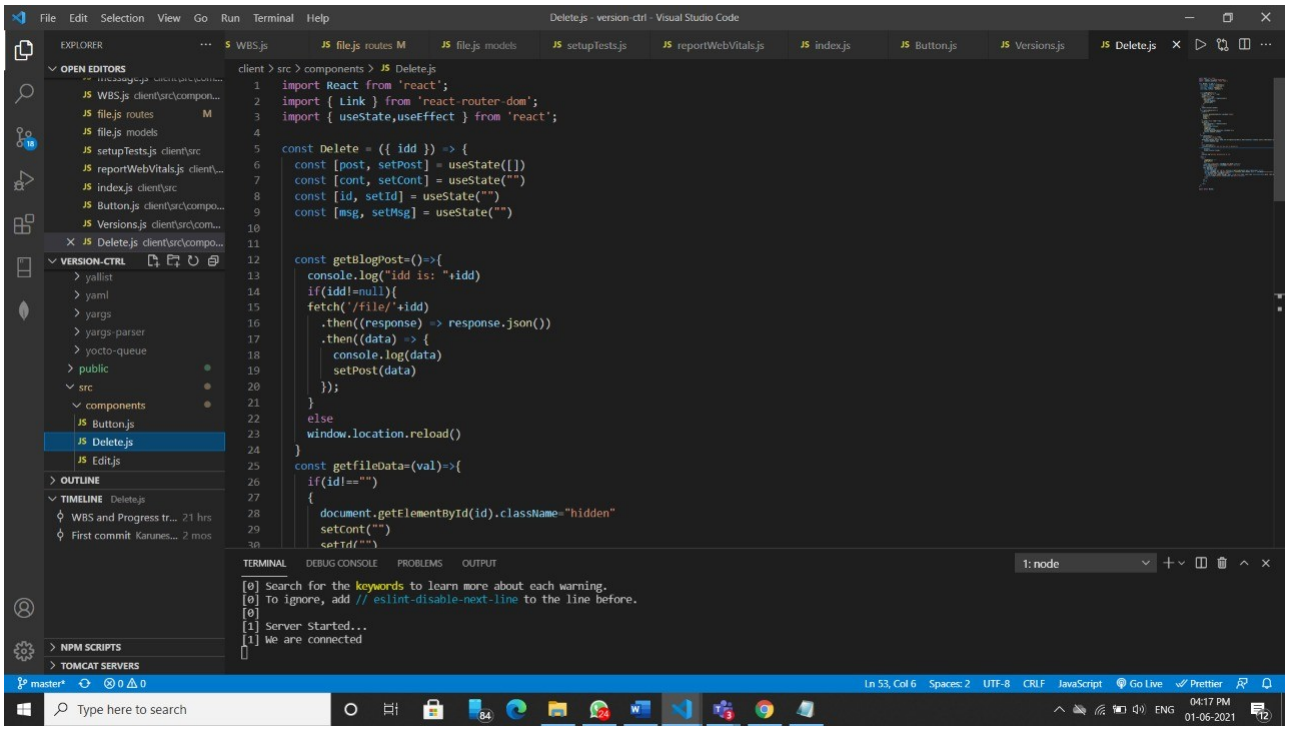
5.1.4Frontend Root



5.1.5Button

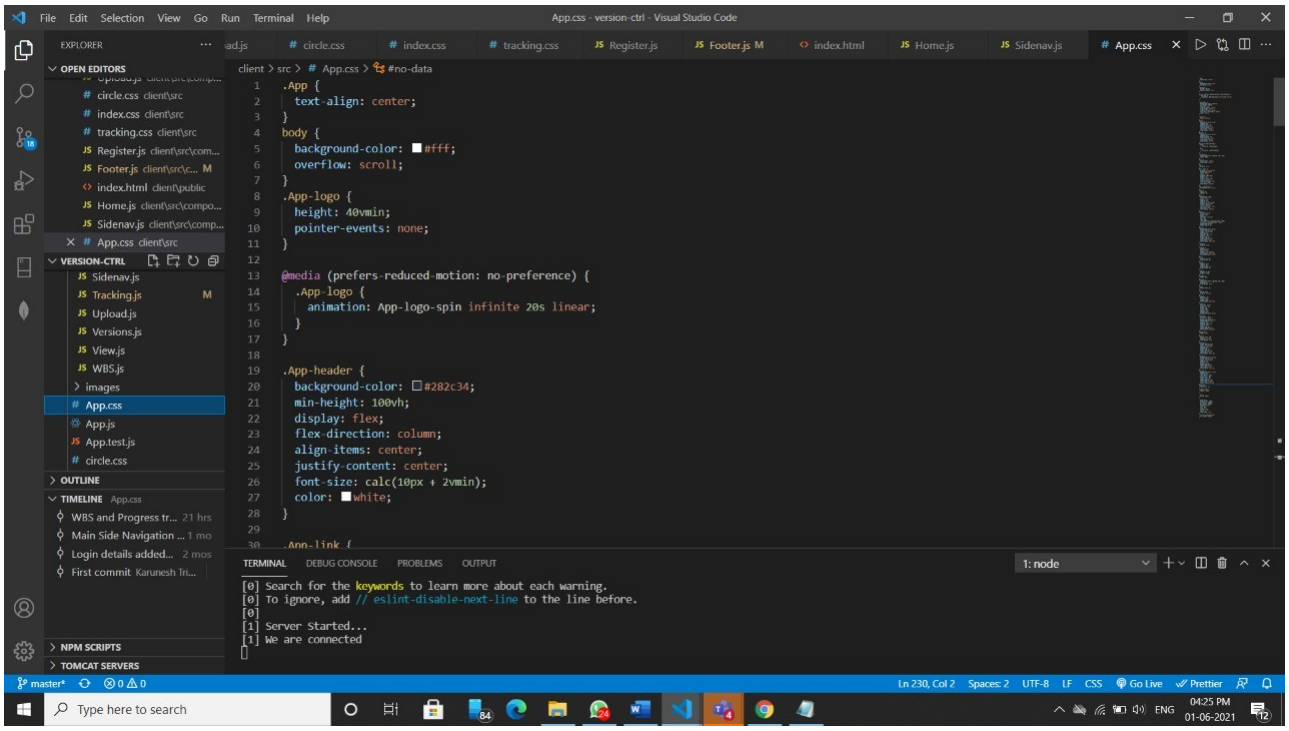


5.1.6Delete File

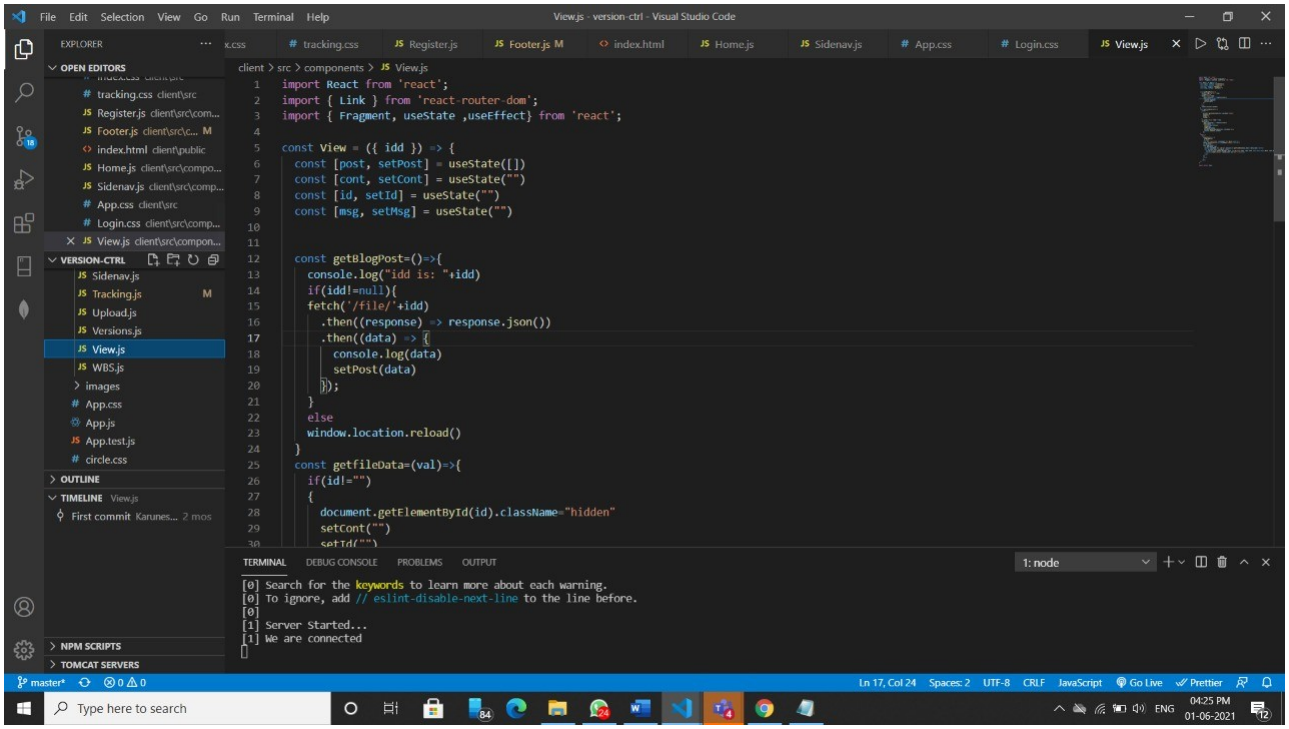


5.2FRONTEND CODE:

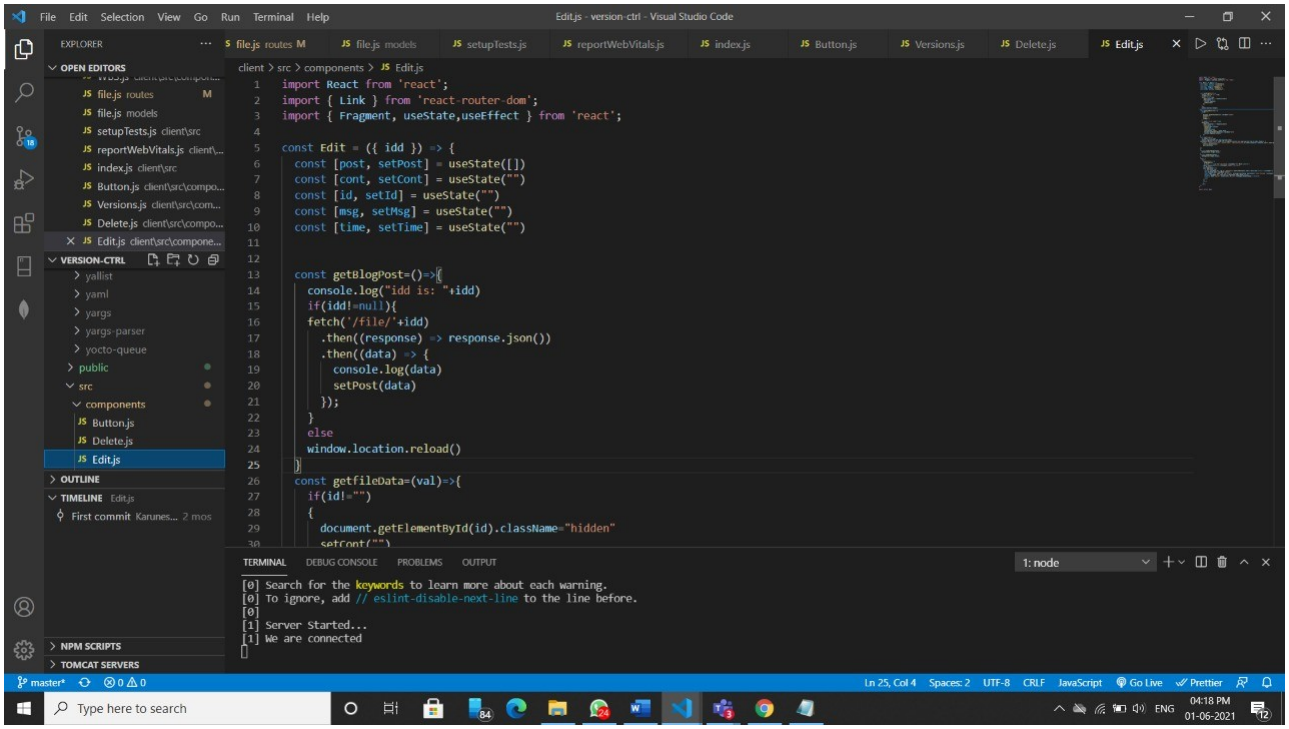
5.2.1App CSS



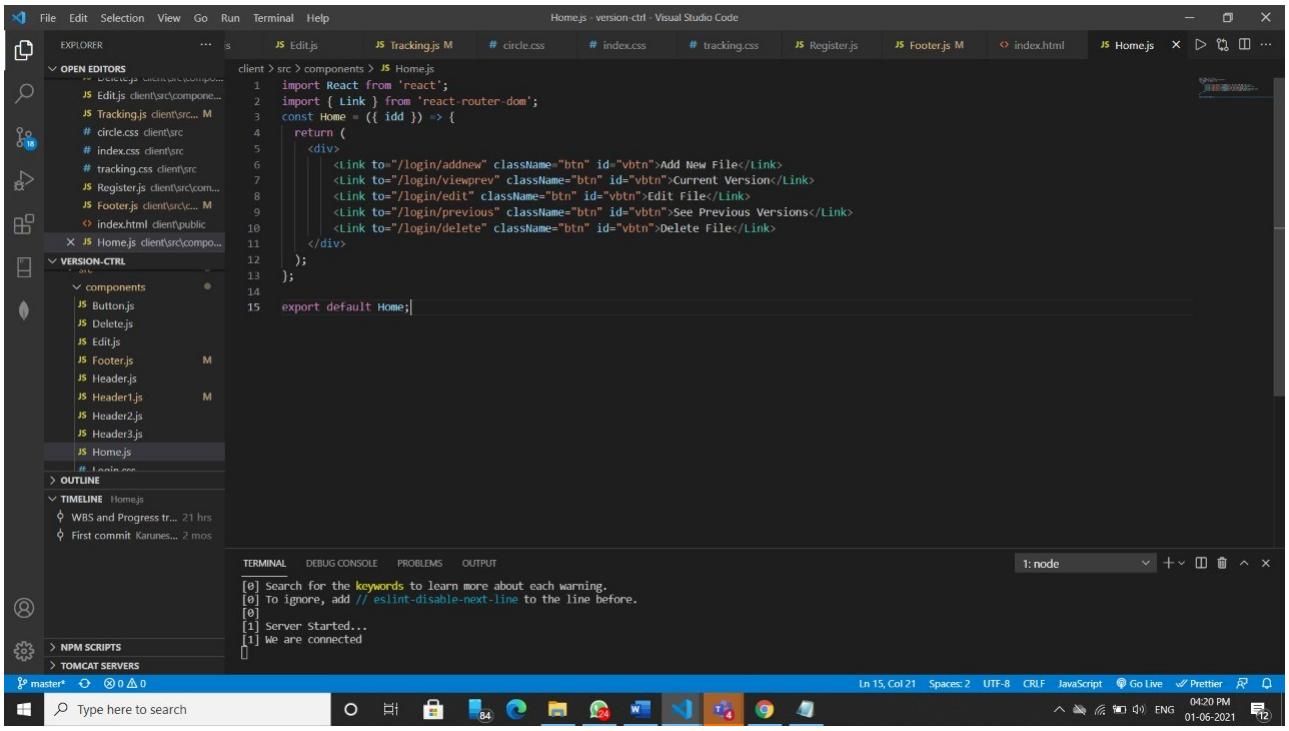
5.2.2Current Version



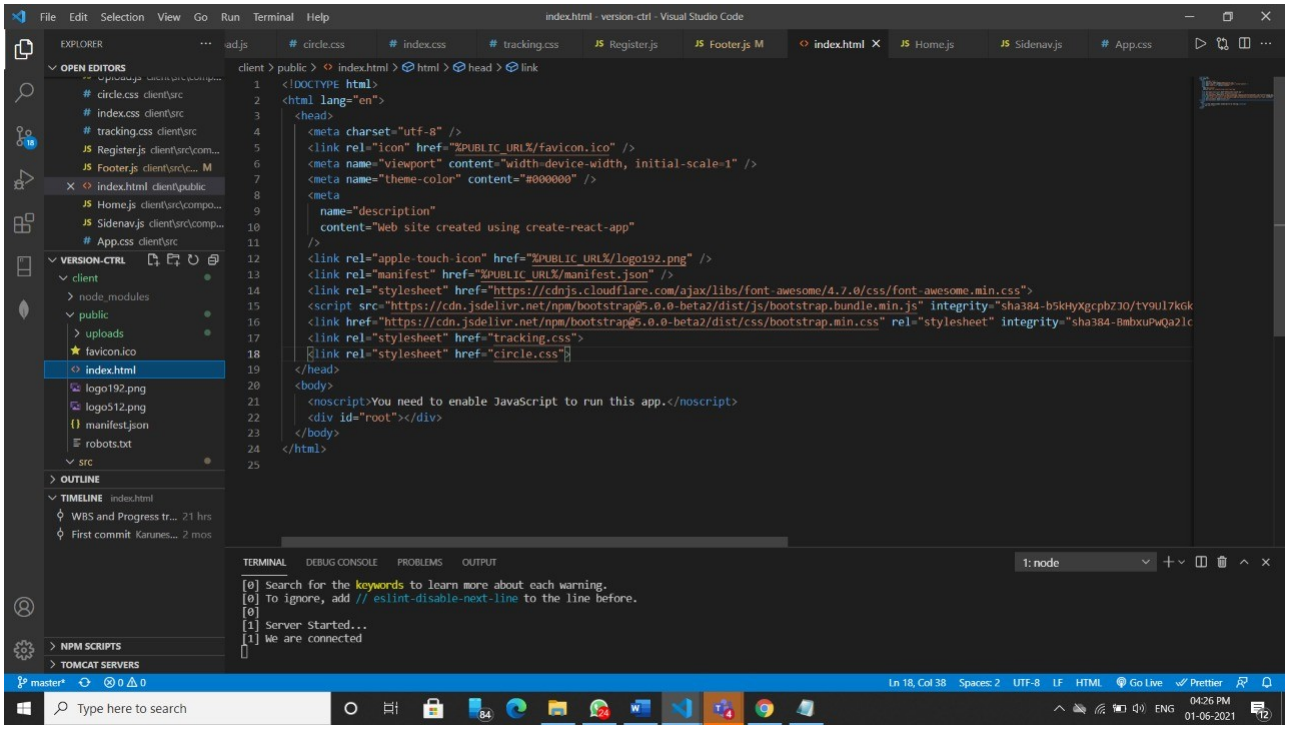
5.2.3Edit File



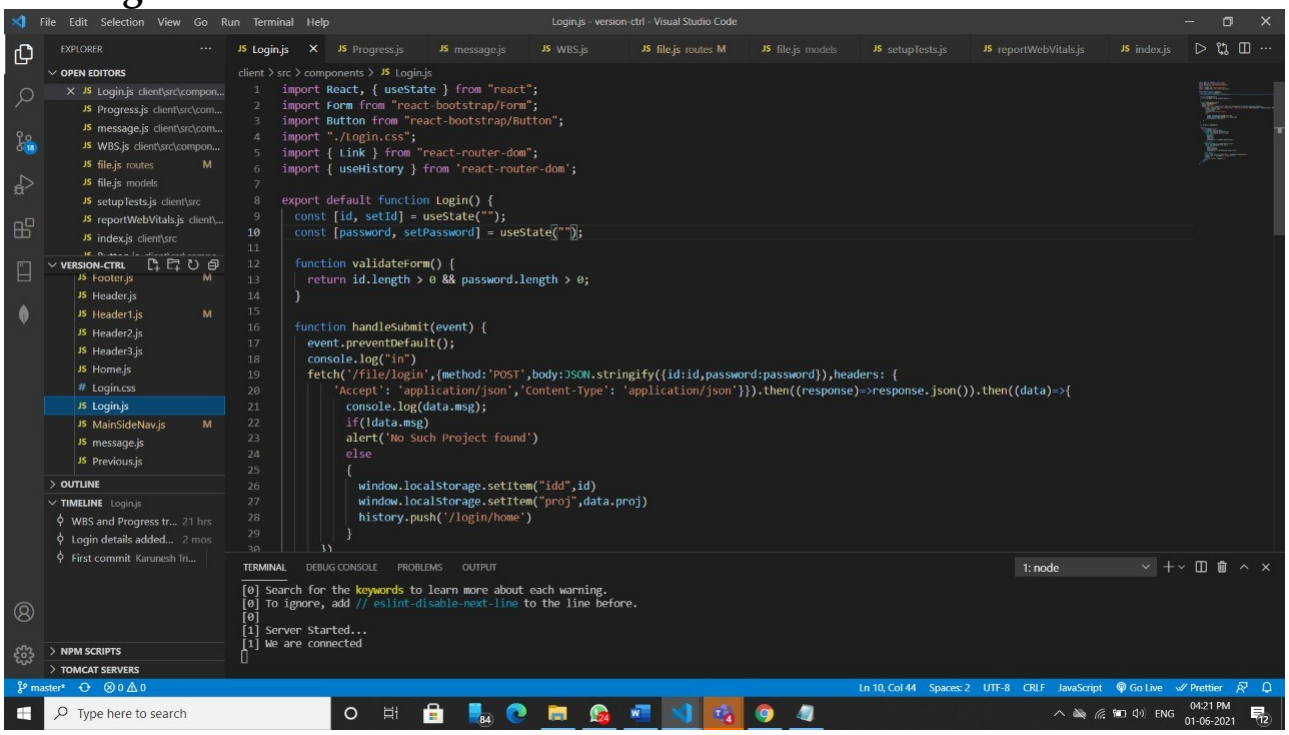
5.2.4Home



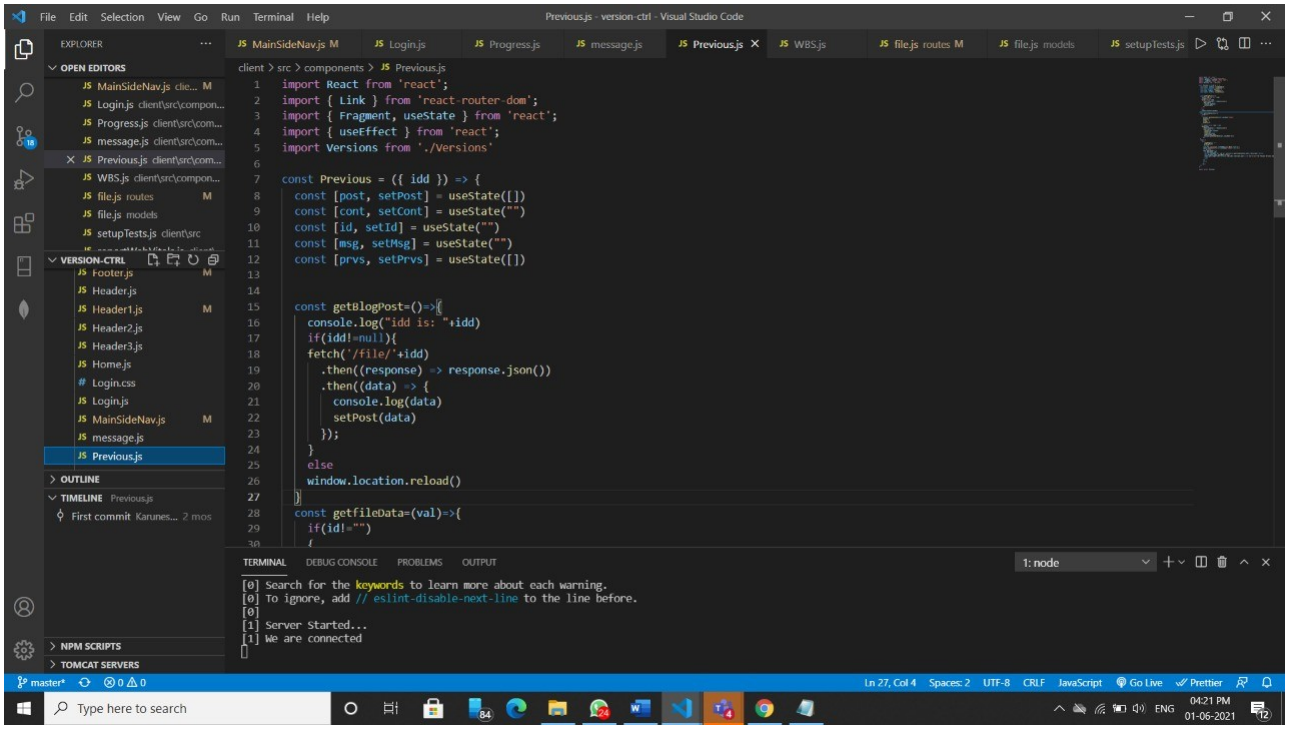
5.2.5HTML Main



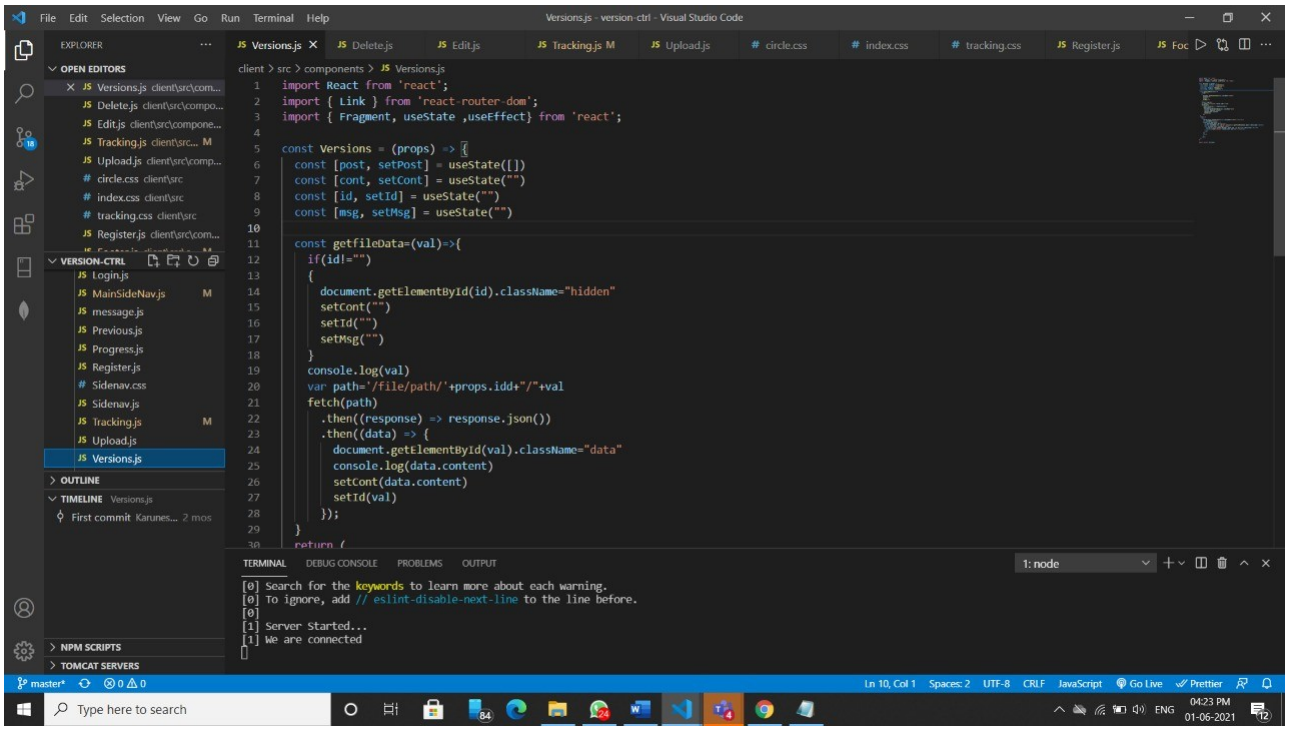
5.2.6Login



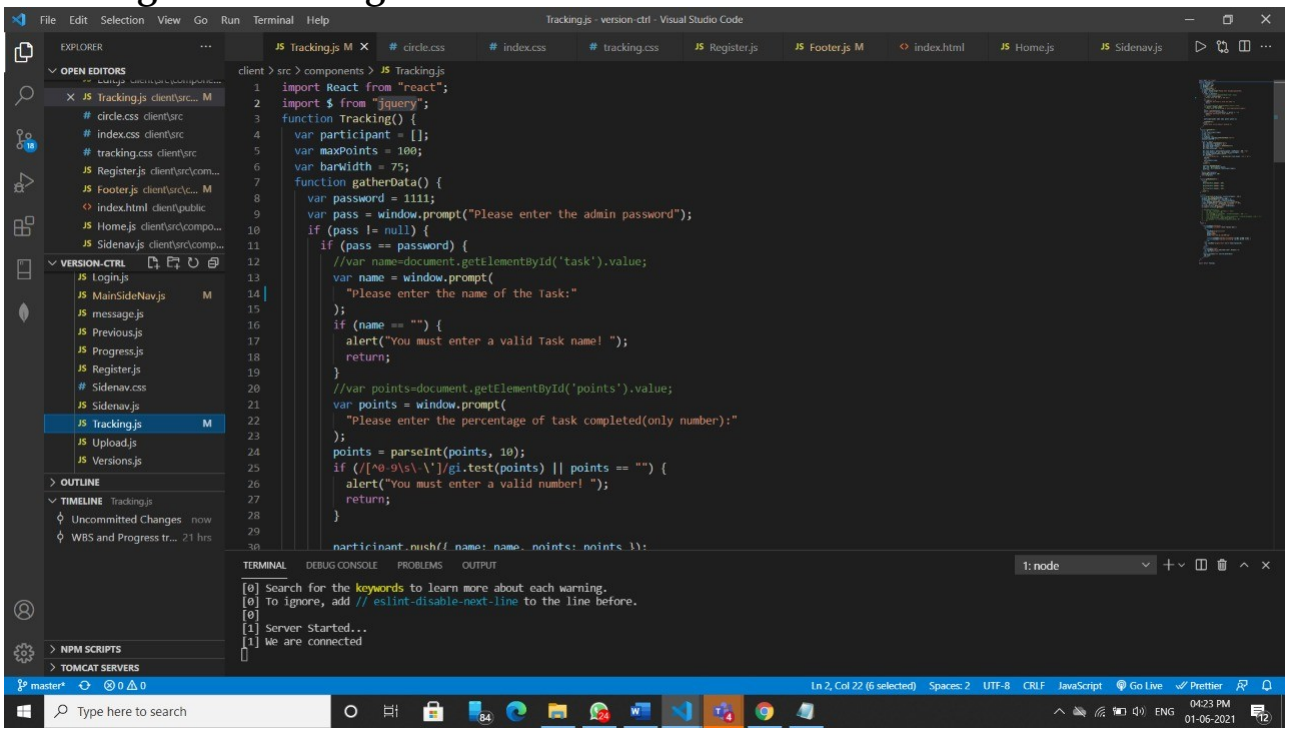
5.2.7Previous Verison



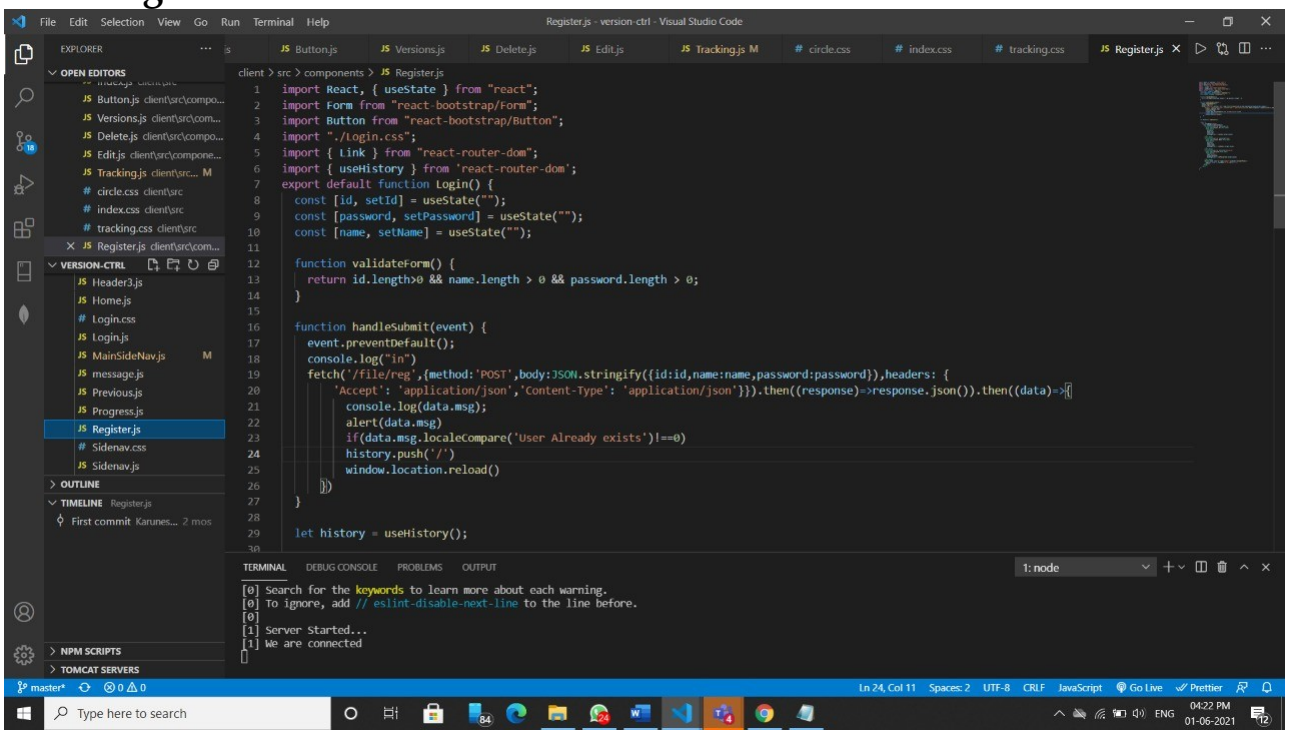
5.2.8Previous Version Fetch



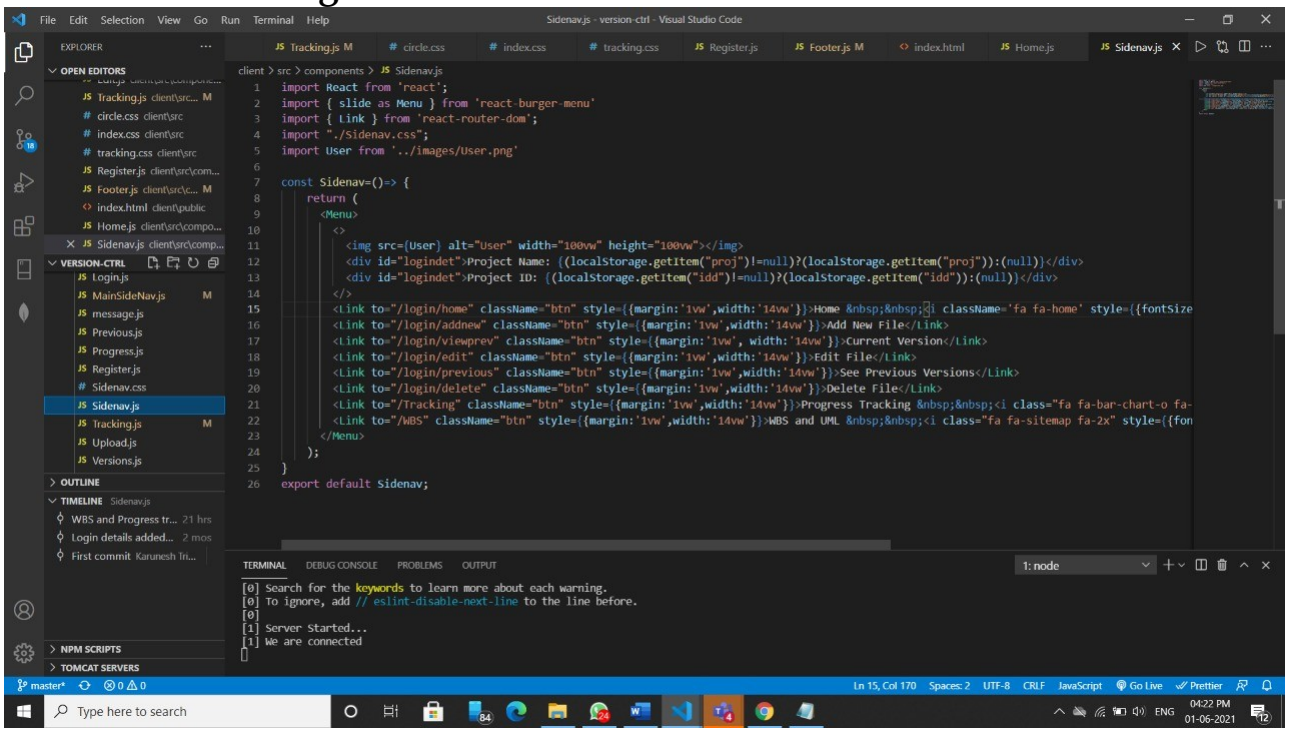
5.2.9Progress Tracking



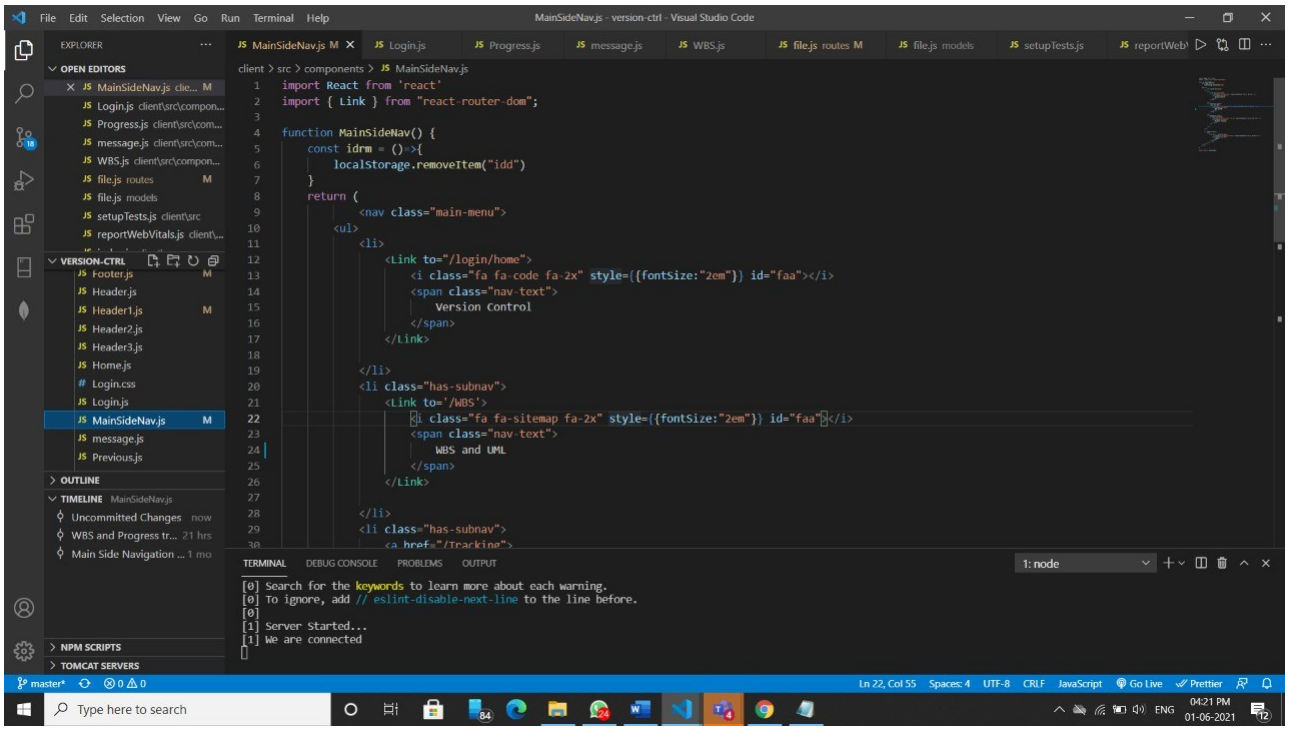
5.2.10Register



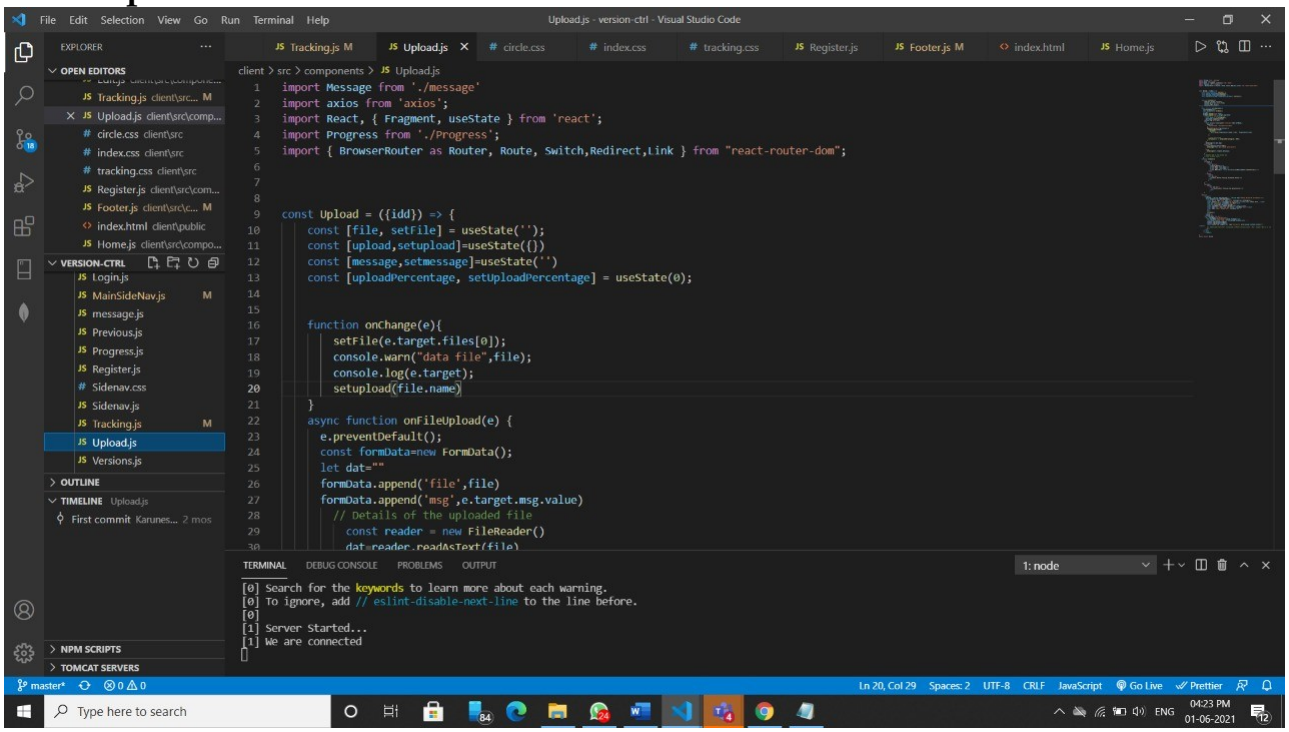
5.2.11Side hamburger menu



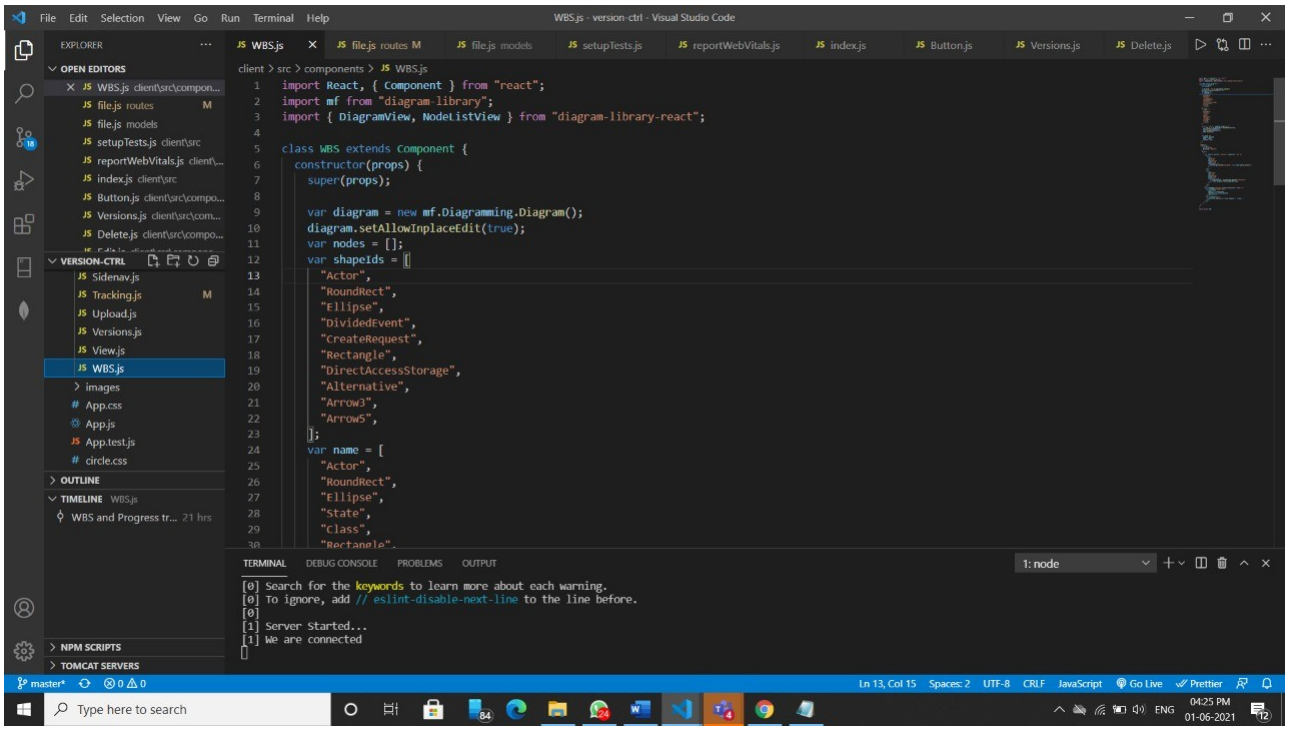
5.2.12Side Nav



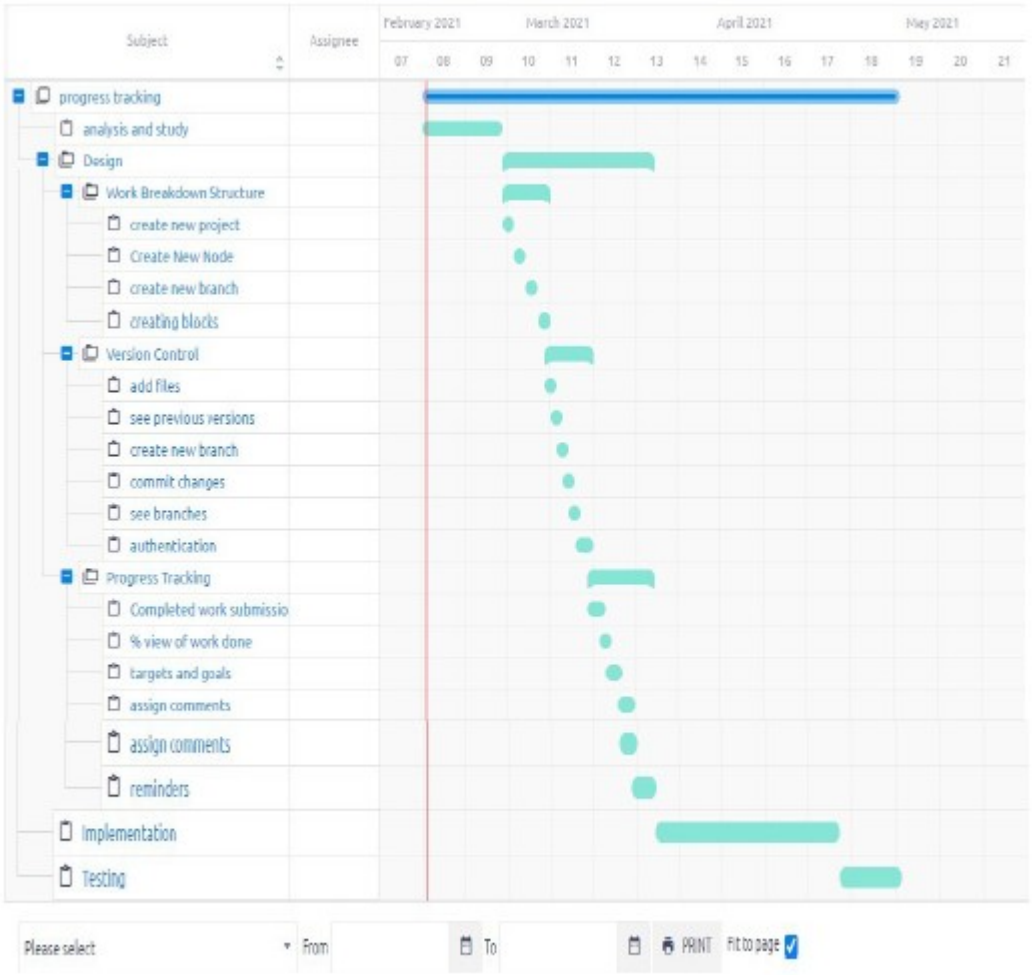
5.2.13Upload



5.2.14WBS and UML

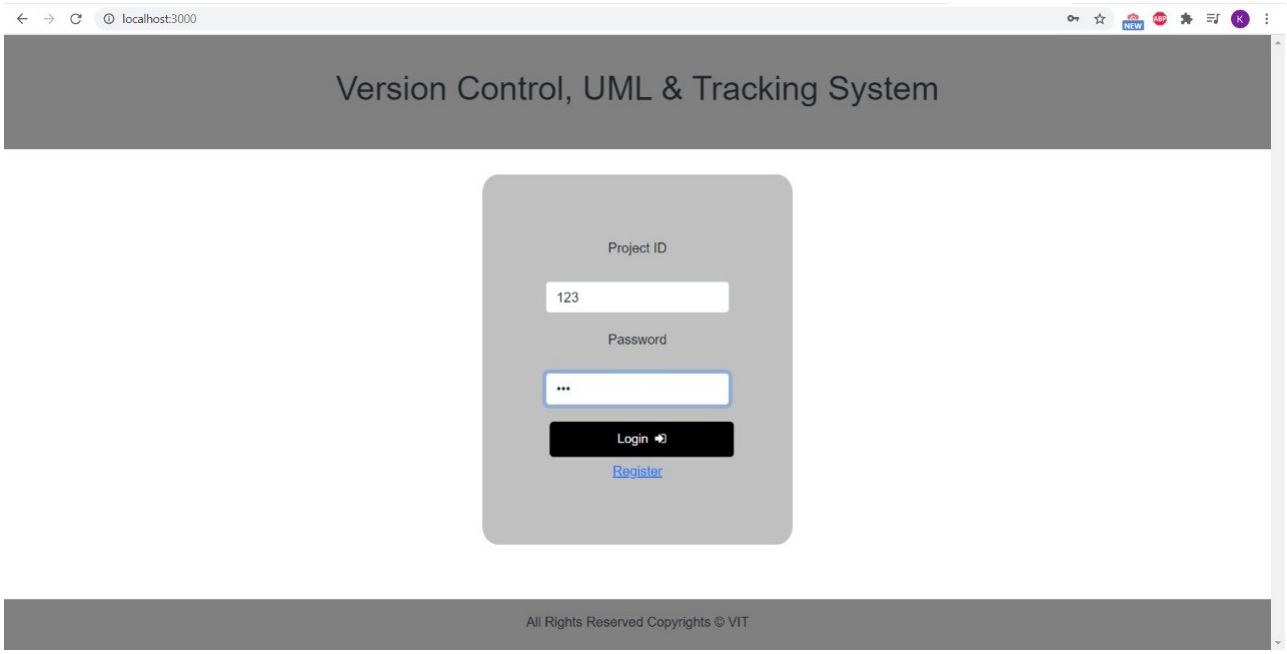


6 SCHEDULE, TASKS AND MILESTONES

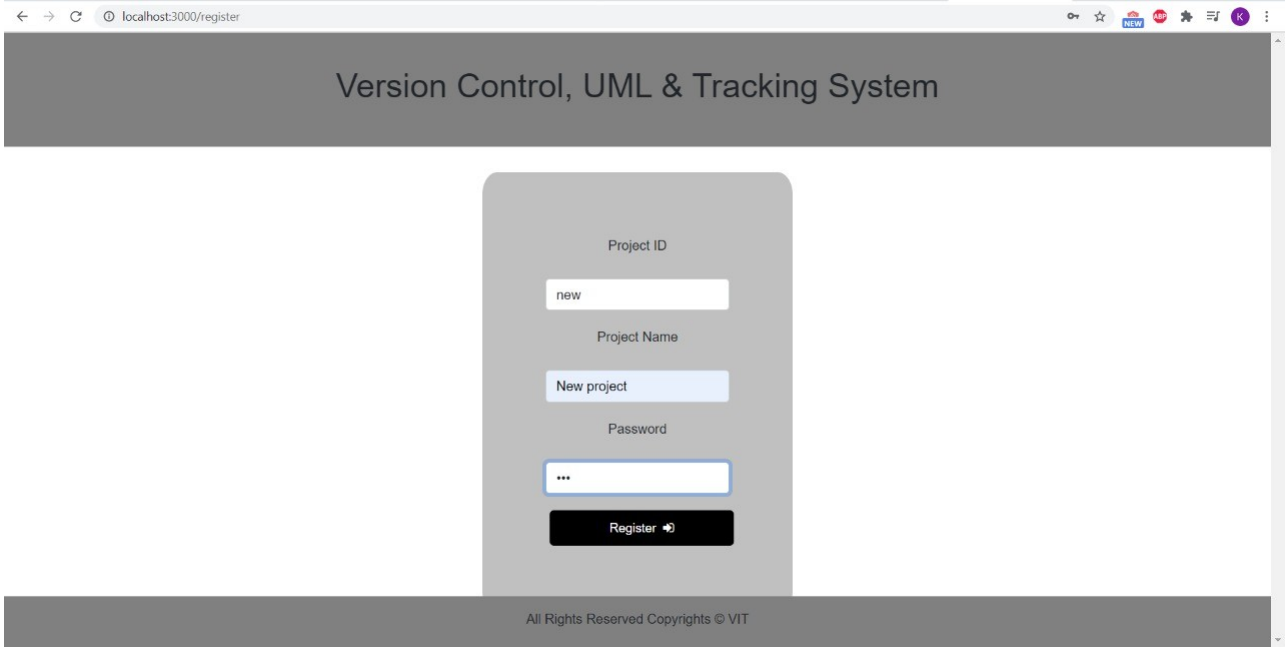


7.Project Demonstration

7.1Log In page

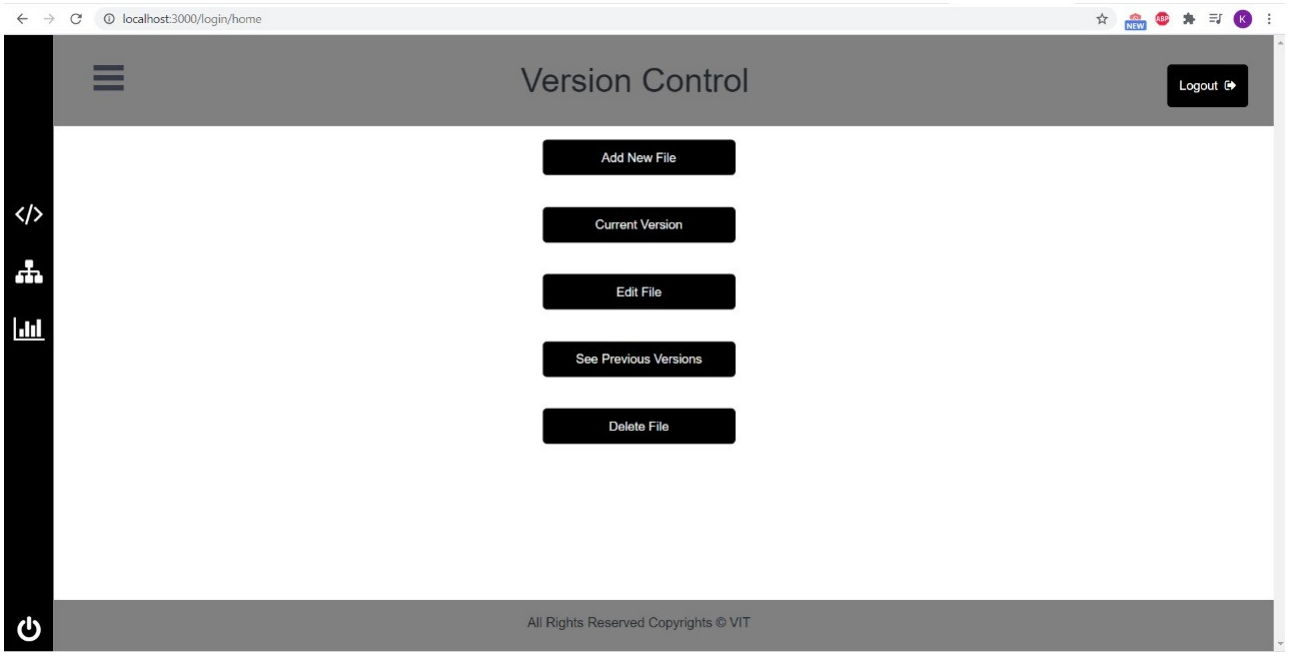


7.2Register

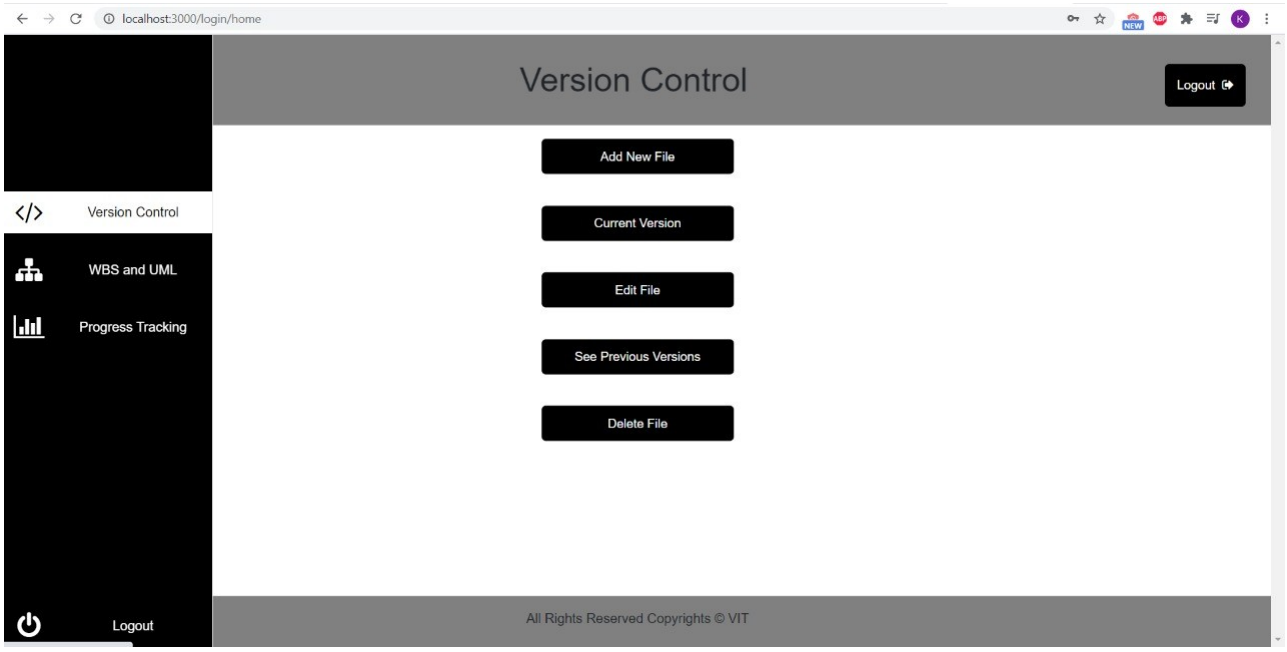


7.3VERSION CONTROL SYSTEM

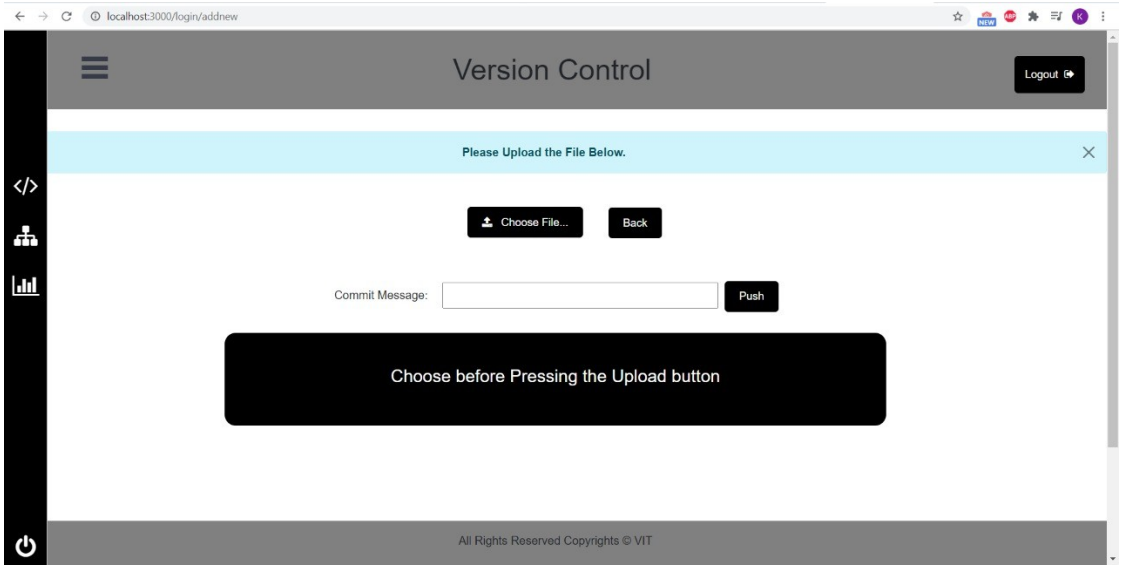
7.3.1HOME PAGE

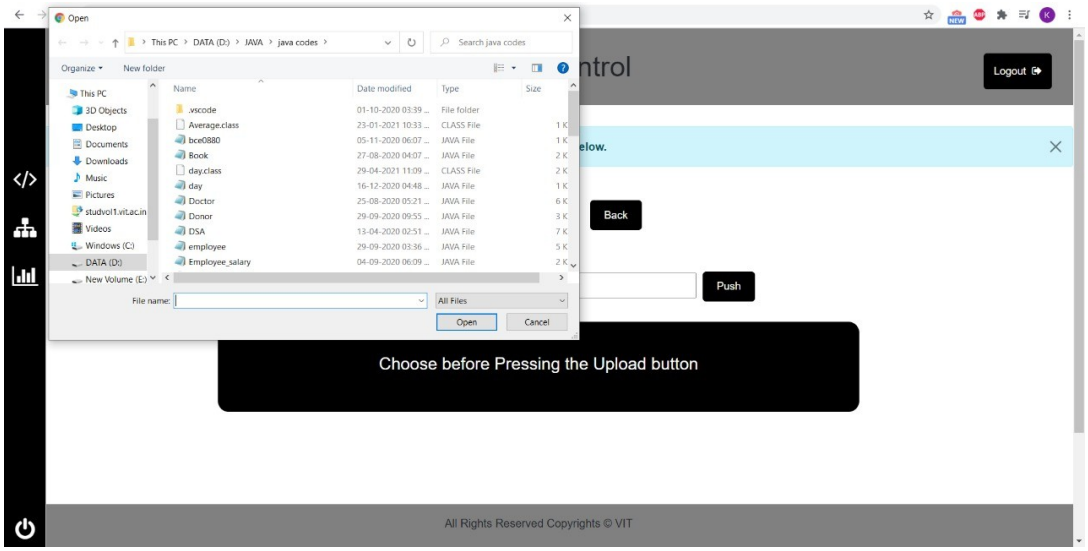


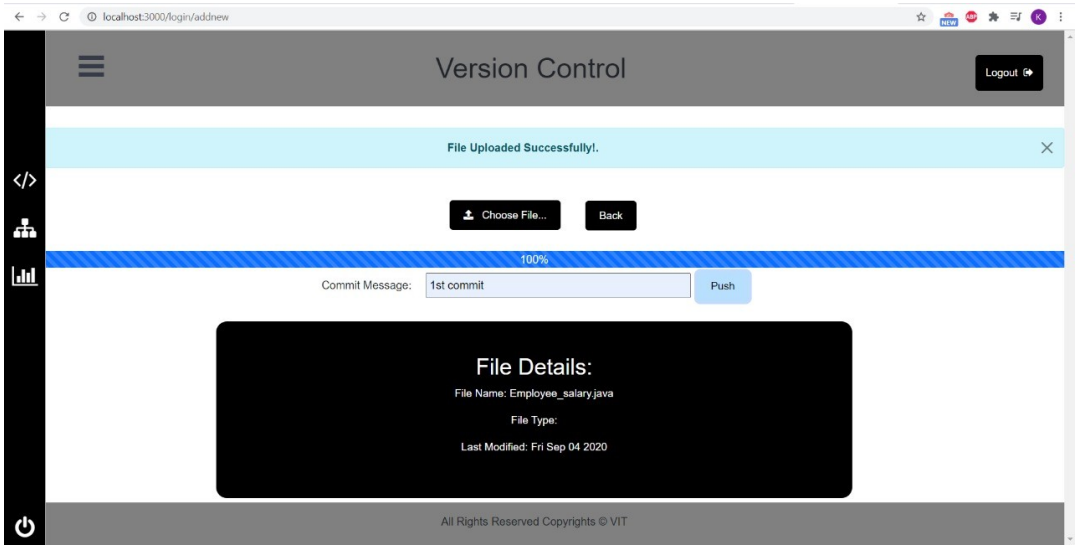
7.3.2NAVIGATION BAR



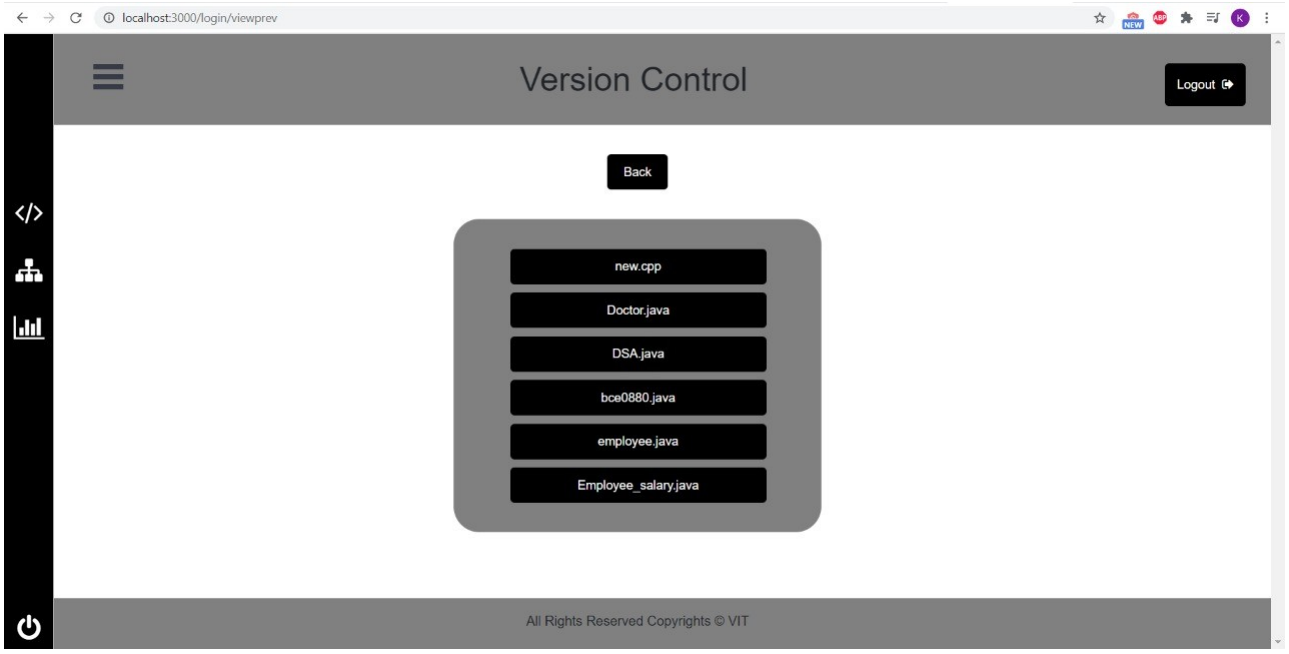
7.3.3ADD NEW FILE

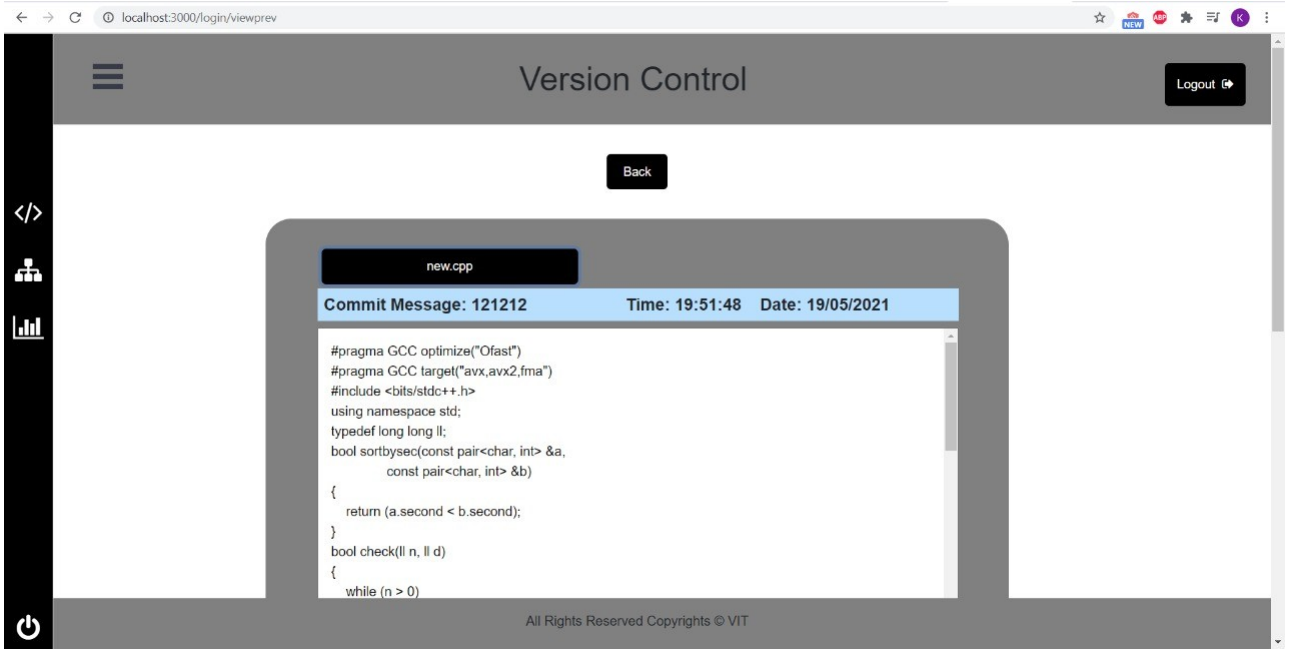



Open
This PC > DATA (D:) > JAVA > java codes >
Search java codes
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
studvol1.vit.ac.in
Videos
Windows (C:)
DATA (D:)
New Volume (E:)
Name
Date modified
Type
Size
.vscode
Average.class
bce0880
Book
day.class
day
Doctor
Donor
DSA
employee
Employee_salary
File name:
All Files
Open
Cancel
ntrol
Logout
elow.
Back
Push
Choose before Pressing the Upload button
All Rights Reserved Copyrights © VIT

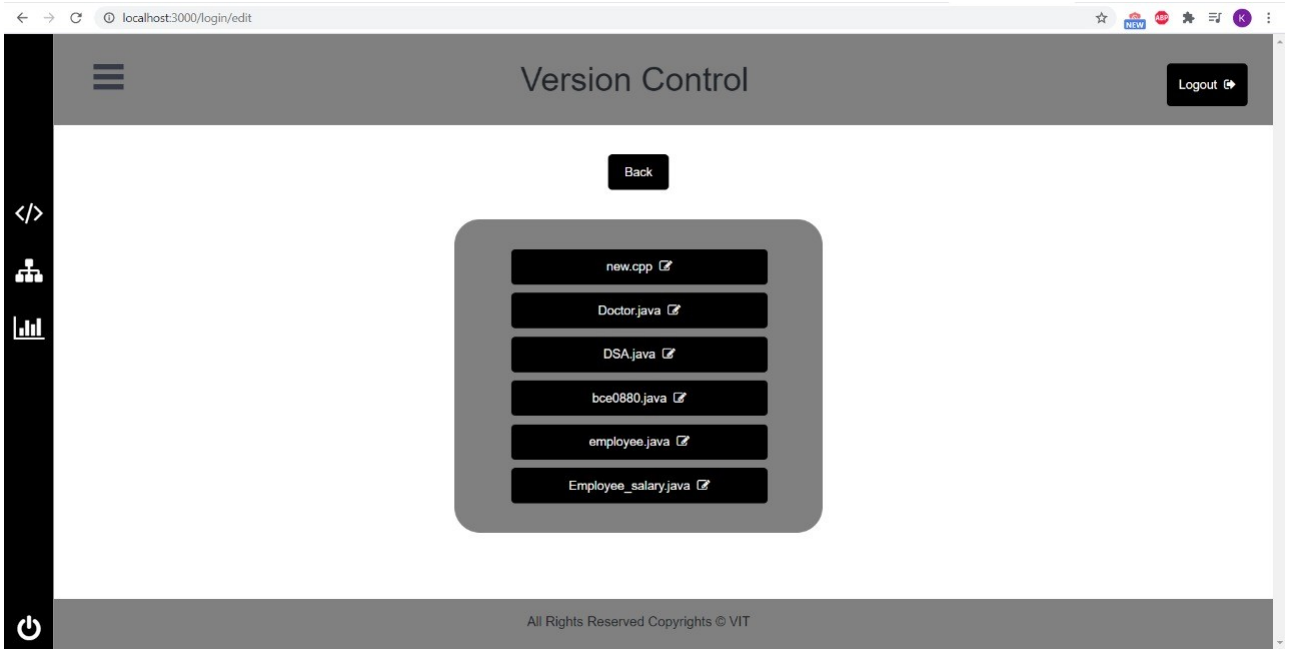

localhost:3000/login/addnew
Version Control
Logout
File Uploaded Successfully!.
Choose File...
Back
100%
Commit Message:
1st commit
Push
File Details:
File Name: Employee_salary.java
File Type:
Last Modified: Fri Sep 04 2020
All Rights Reserved Copyrights © VIT

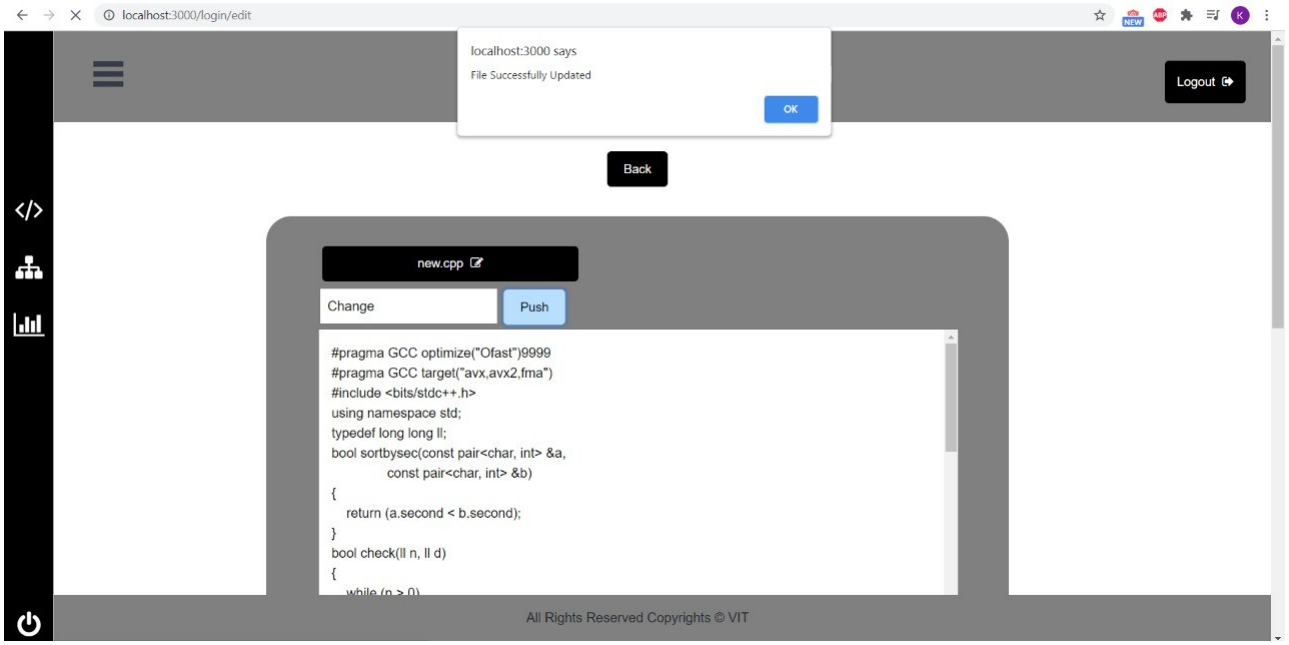
7.3.4SEE CURRENT VERSION



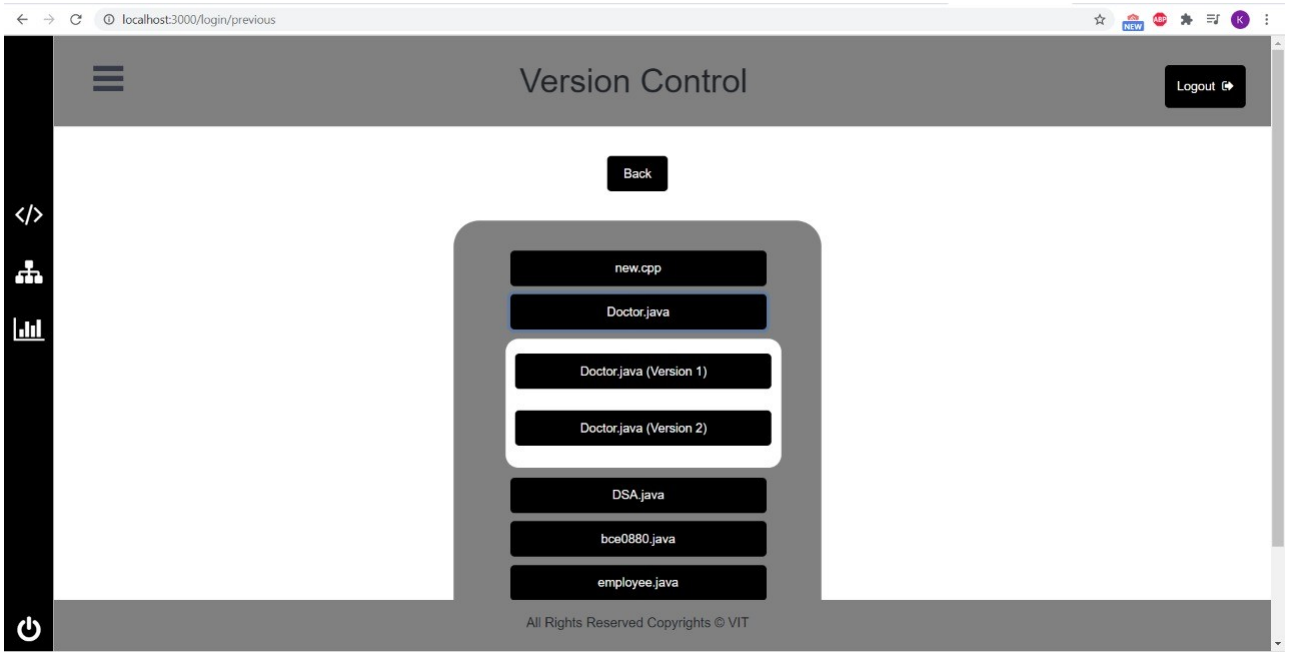


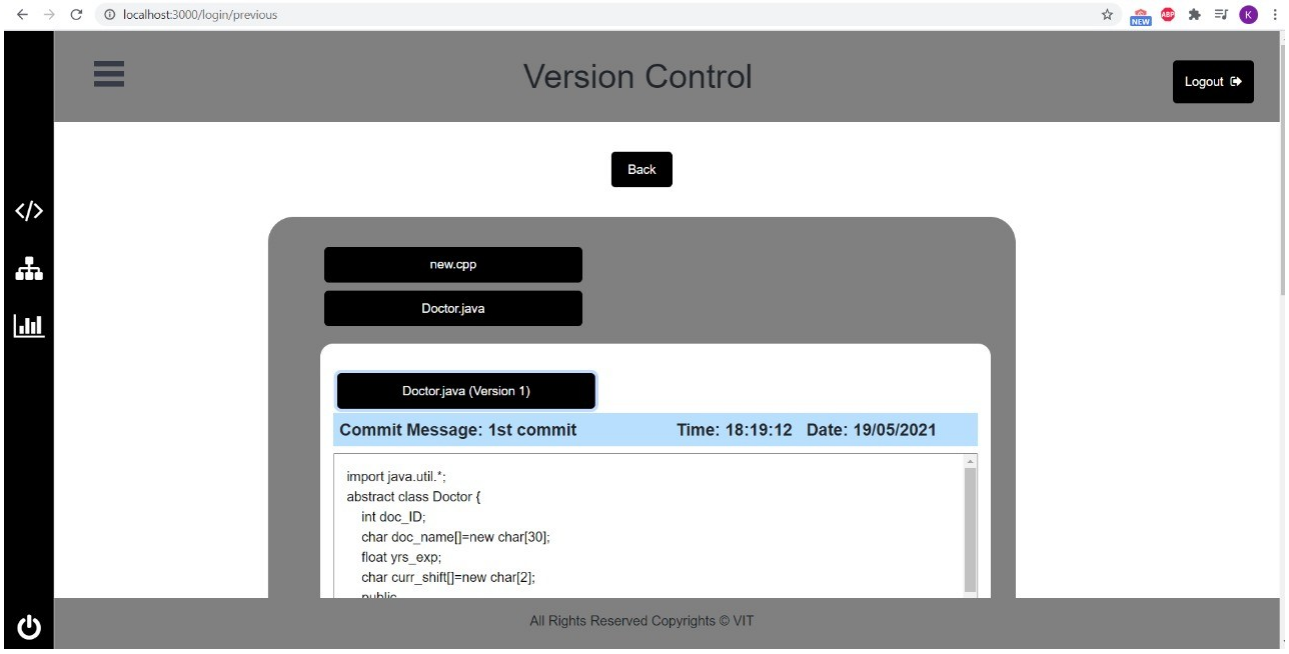
7.3.5EDIT EXISTING FILE



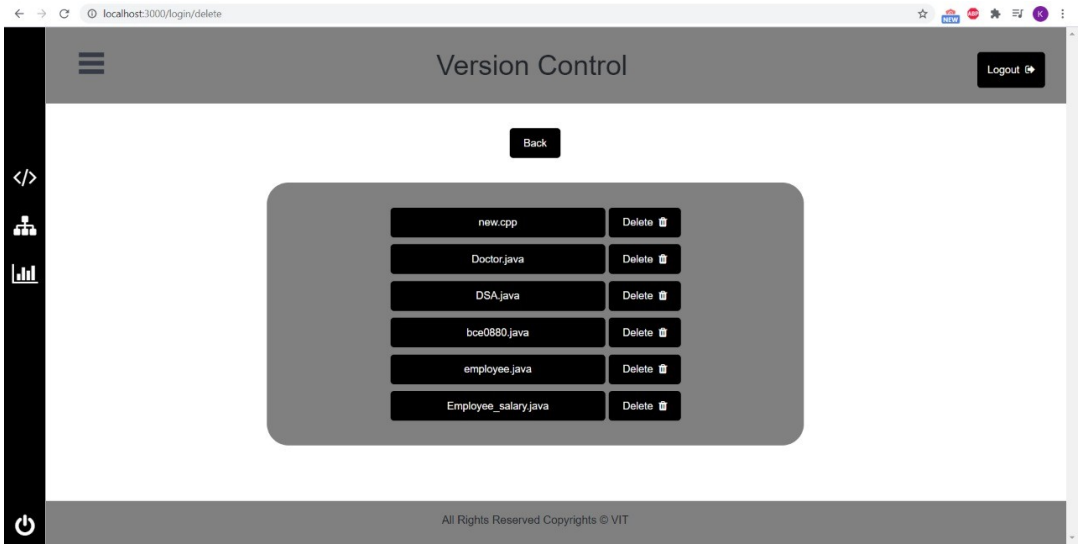


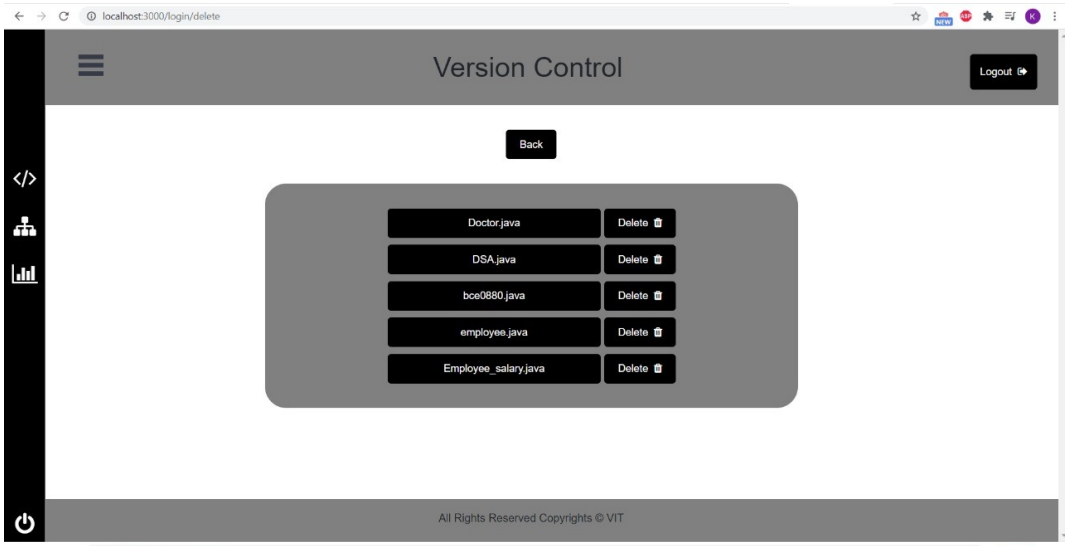
7.3.6SEE PREVIOUS VERSION

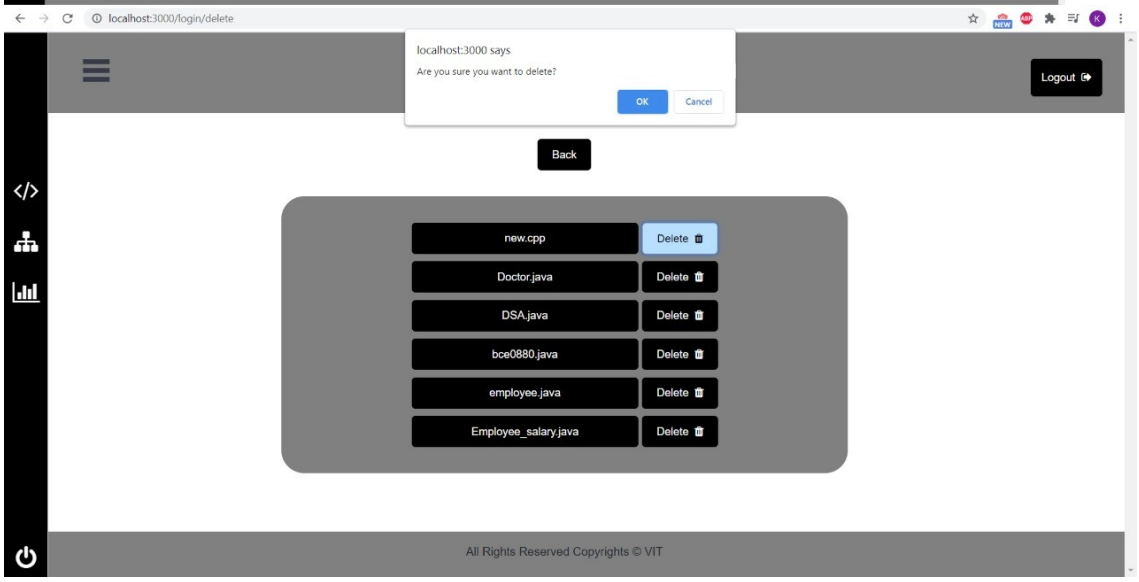




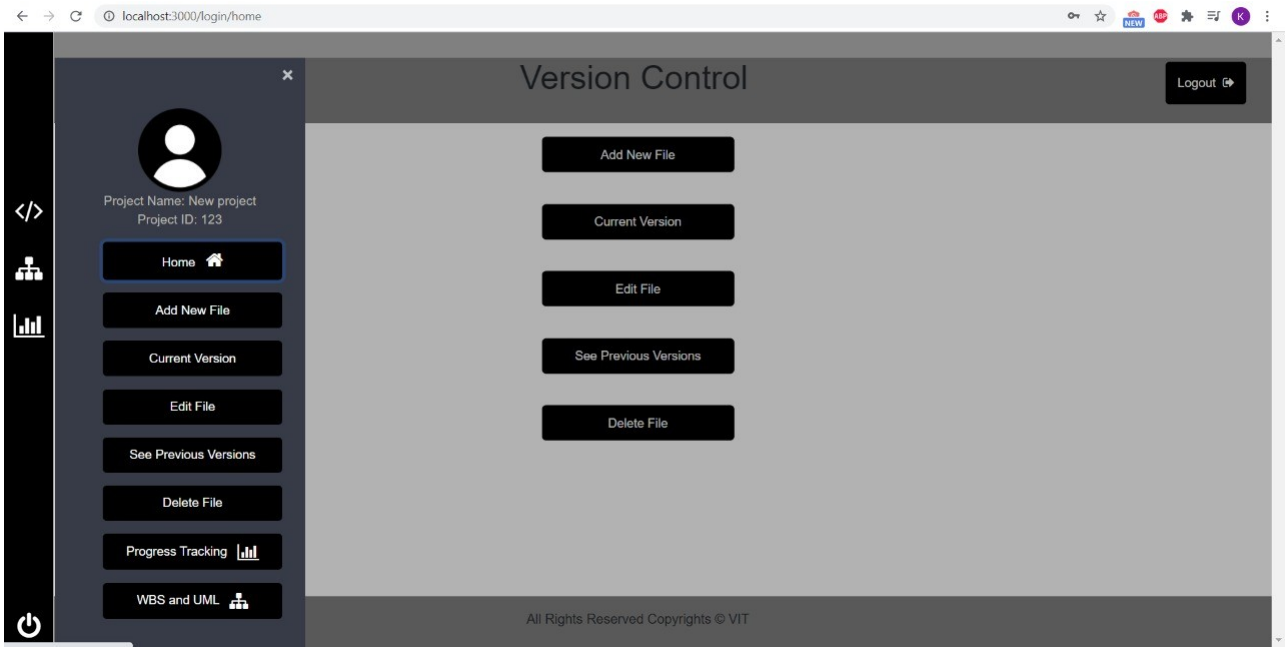
7.3.7DELETE FILE





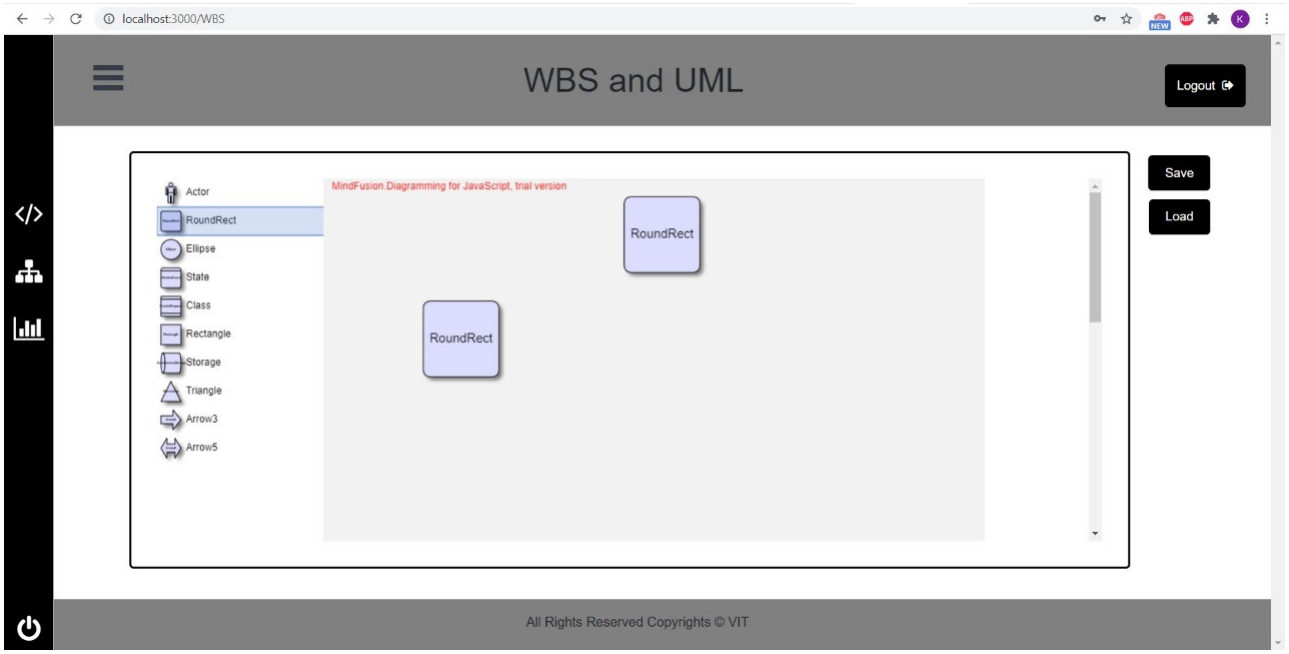


7.3.8MENU NAVIGATION BAR

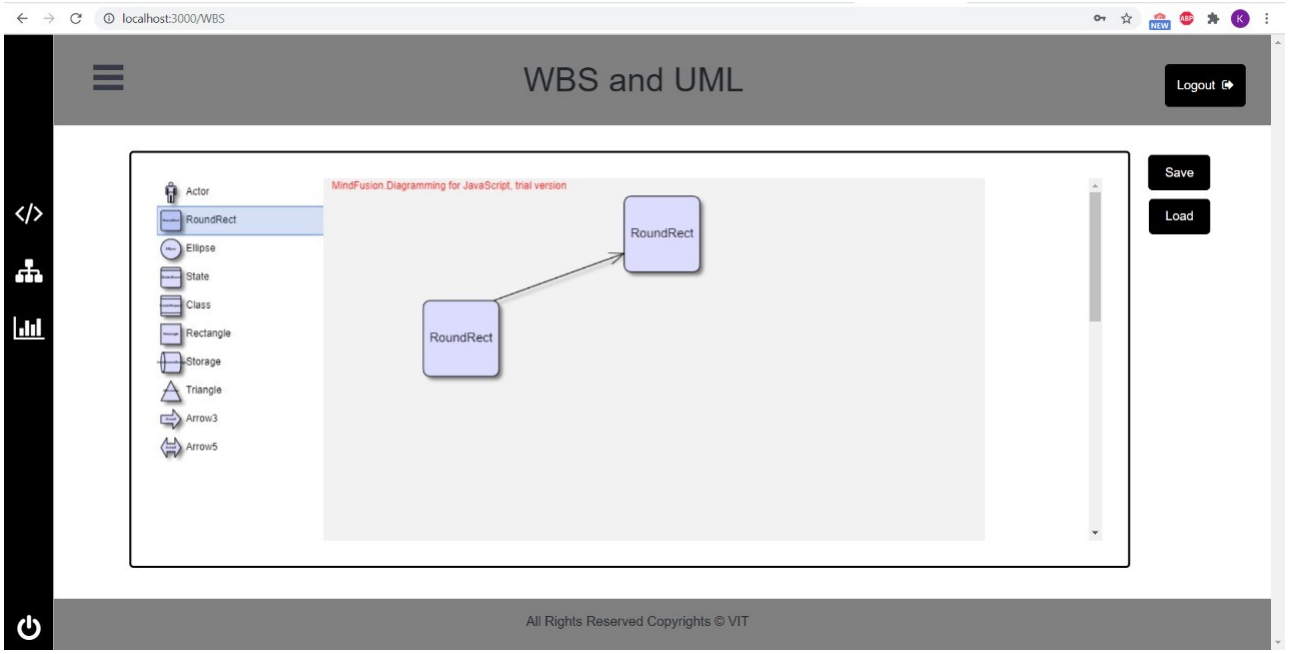


7.4WBS and UML Generation

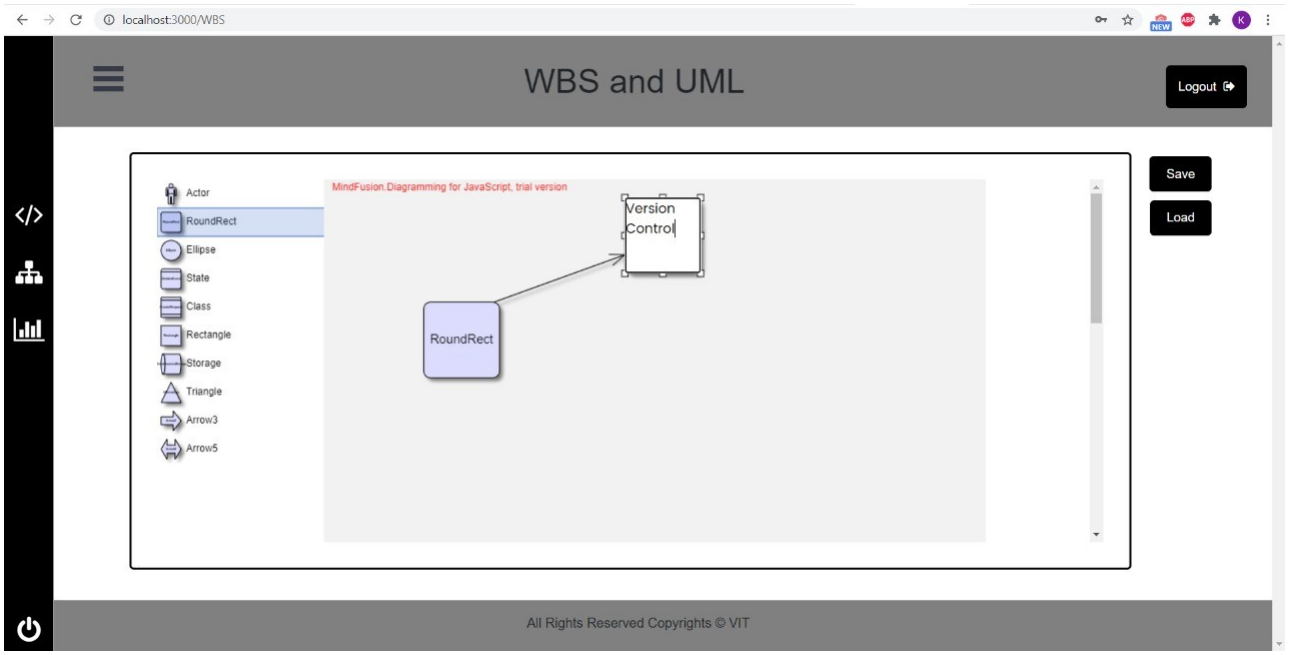
7.4.1Drag and drop shapes



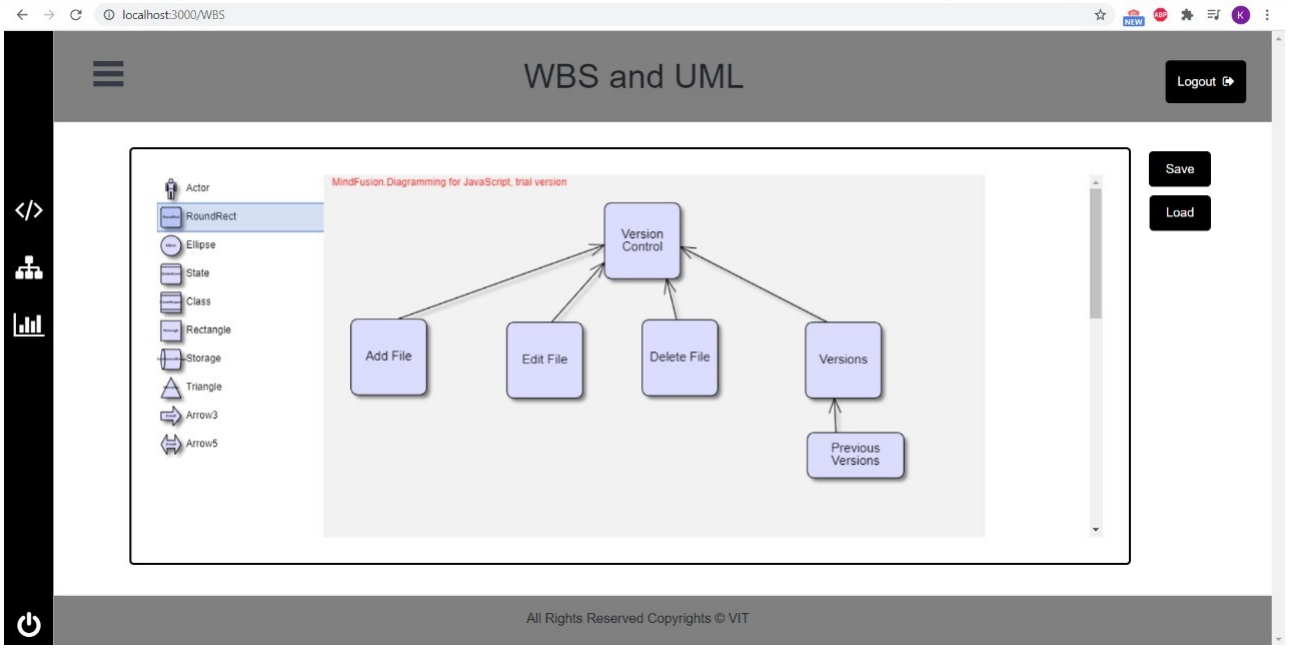
7.4.2Connect shapes



7.4.3Rename

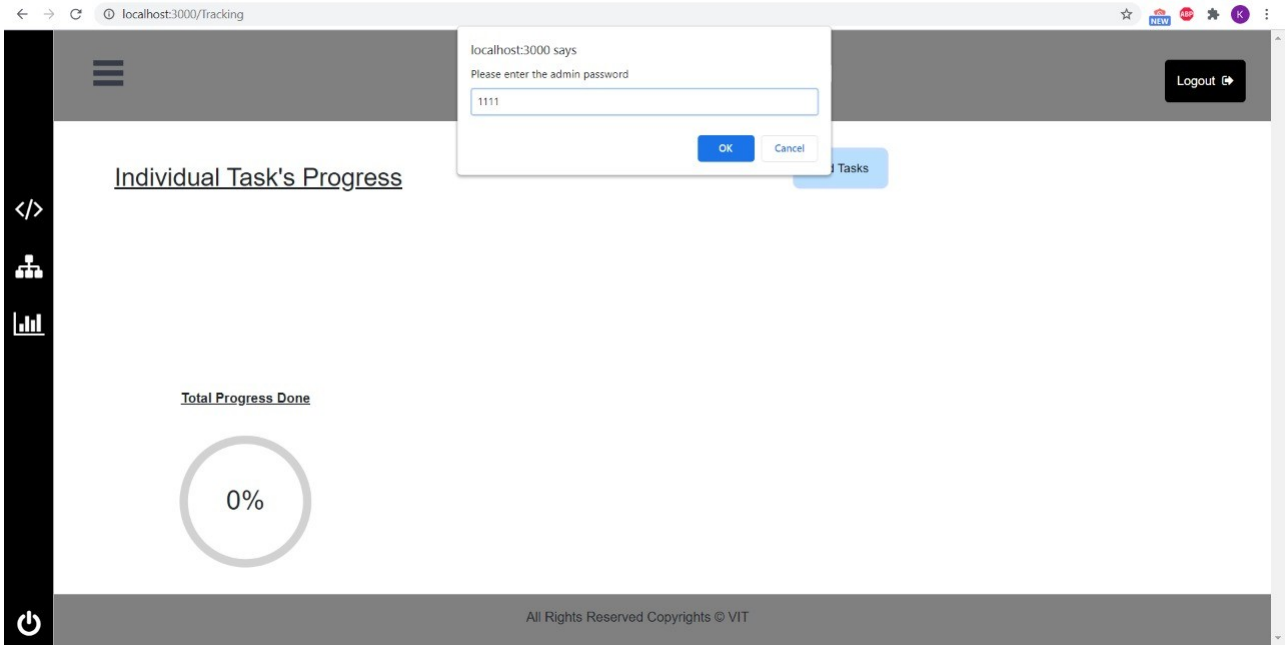


7.4.4Final

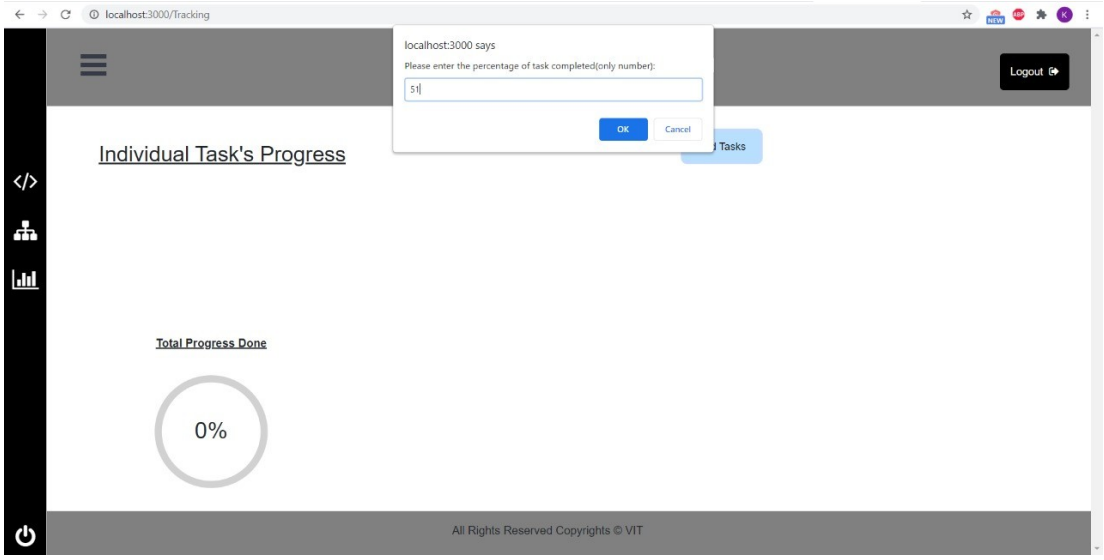


7.5Progress Tracking

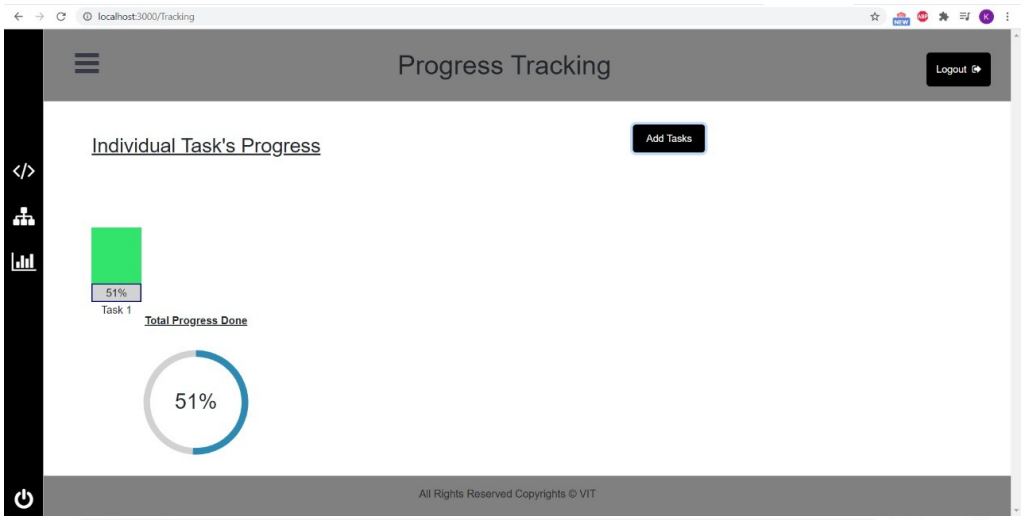
7.5.1Admin password required to access tracking record

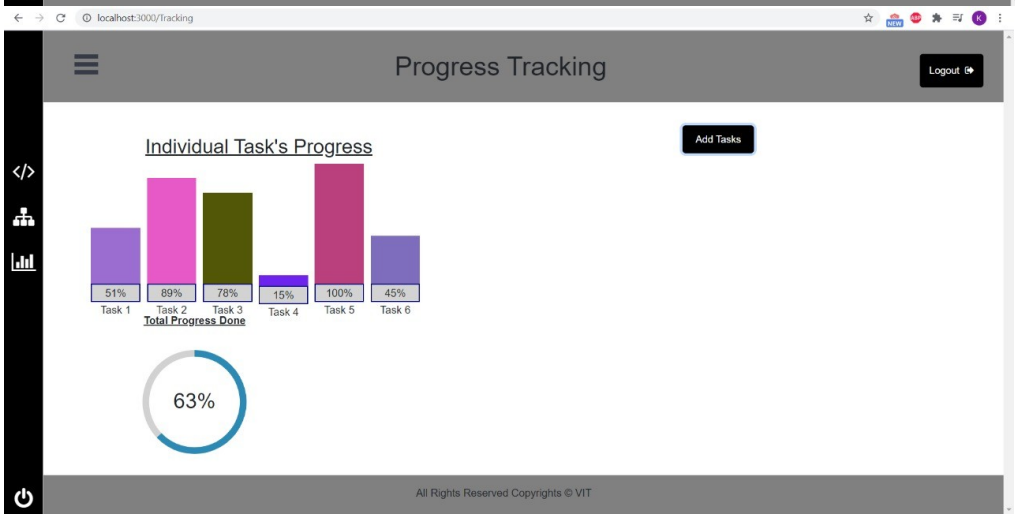


7.5.2Enter data



7.5.3Tracking





List of Tables

LOGIN PAGE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_1	Check login with Unregistered id	Project ID:-456 Password:-456	Message:- No such Project found	Message:- No such Project found	Pass
VC_2	Login with registered ID wrong password	Project ID-123 Password-456	Message:- No such Project found	Message:- No such Project found	Pass
VC_3	Login with valid credentials	Project ID-123 Password-123	Redirect to Home page	Redirected to Home page	Pass
VC_4	Check Register hyperlink	Click on Register	Redirect to Register page	Redirected to Register page	Pass

REGISTER PAGE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_5	Register without entering all the credentials	Project ID:- Project Name:- Password:-	Register button disabled	Register button disabled	Pass
VC_6	Register without entering any one of the credentials	Project ID:-new Project Name:-new Password:-	Register button disabled	Register button disabled	Pass
VC_7	Register with valid credentials	Project ID:-new Project Name:-new Password:-new	Redirect to Login page	Redirected to Login page	Pass
VC_8	Register with existing credentials	Project ID:-new Project Name:-new Password:-new	Message:- User Already exists	Message:- User Already exists	Pass

HOME PAGE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_9	Check Add New File button	Click on add new file button	Redirect to Add new file page	Redirected to Add new file page	Pass
VC_10	Check Current Version button	Click on current version	Redirect to current version page	Redirected to current version page	Pass
VC_11	Check See Previous Versions button	Click on See Previous Versions button	Redirect to See Previous Versions page	Redirected to See Previous Versions page	Pass
VC_12	Check Edit File button	Click on Edit File button	Redirect to Edit file page	Redirected to Edit file page	Pass
VC_13	Check Delete File button	Click on Delete File button	Redirect to Delete file page	Redirected to Delete file page	Pass
VC_14	Check Logout button	Click on Logout button	Redirect to Login page	Redirected to Login page	Pass
VC_15	Check Hamburger menu button	Click on Hamburger button	Open menu	Menu opened	Pass
VC_16	Check Side Menu for progress tracking	Click on Progress	Redirect to Progress tracking	Redirected to Progress tracking	Pass

	button	tracking button	page	page	
VC_17	Check Side Menu for WBS button	Click on WBS button	Redirect to WBS page	Redirected to WBS page	Pass

ADD NEW FILE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_18	Check Choose File button	Click on choose file button and enter file to be selected	File selected	File selected	Pass
VC_19	Check Text input commit message	First commit	Text Box:- First commit	Text Box:- First commit	Pass
VC_20	Check Choose File button	Click on choose file button and enter file to be selected	File selected	File selected	Pass
VC_21	Check push button	Click on push button	File and commit message upload and success message displayed	File and commit message uploaded and success message displayed	Pass
VC_22	Check Back button	Click on Back button	Redirect to home page	Redirected to home page	Pass

CURRENT VERSION PAGE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_23	Open file Donor.java in current version	Click on Donor.java button	Donor.java file should open	Donor.java file opened	Pass
VC_24	Open file new.cpp in current version	Click on new.cpp button	new.cpp file should open	new.cpp file opened	Pass
VC_25	Check Back button	Click on Back button	Redirect to home page	Redirected to home page	Pass

PREVIOUS VERSIONS

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_26	Check Donor.java button	Click on Donor.java button	Open previous versions of Donor.java	Opened previous versions of Donor.java	Pass
VC_27	Open file Donor.java(1)	Click on Donor.java(1) button	Previous version Donor.java(1) file should open	Previous version Donor.java(1) file opened	Pass
VC_28	Open file Donor.java(2)	Click on Donor.java(2) button	Previous version Donor.java(2) file should open	Previous version Donor.java(2) file opened	Pass
VC_29	Check new.cpp button	Click on new.cpp button	Open previous versions of new.cpp	Opened previous versions of new.cpp	Pass
VC_30	Check Back button	Click on Back button	Redirect to home page	Redirected to home page	Pass

EDIT FILE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_31	Check Donor.java button	Click on Donor.java button	Open Donor.java file for editing	Opened Donor.java for editing	Pass
VC_32	Check Text input commit message	New commit	Text Box:- New commit	Text Box:- New commit	Pass
VC_33	Edit new.cpp	Click on new.cpp button	Open new.cpp file for editing	Opened new.cpp file for editing	Pass
VC_34	Check push button	Click on push button	File and commit message upload and success message displayed. Add Current version of file to previous version	File and commit message uploaded and success message displayed. Added Current version of file to previous version	Pass
VC_35	Check Back button	Click on Back button	Redirect to home page	Redirected to home page	Pass

DELETE FILE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
VC_36	Check Donor.java button	Click on Donor.java button	Open Donor.java file for viewing	Opened Donor.java for viewing	Pass
VC_37	Check new.cpp button	Click on new.cpp button	Open new.cpp file for viewing	Opened new.cpp for viewing	Pass
VC_38	Check Delete button to delete Donor.java button	Click on Delete button	Ask confirmation: Are you sure you want to delete?	Asked confirmation: Are you sure you want to delete?	Pass
VC_39	Delete file by clicking OK in confirmation box	Click on OK button of confirmation box	Delete Donor.java file along with it Previous versions	Donor.java file deleted along with it Previous versions	Pass
VC_40	Cancel Deletion by clicking cancel in confirmation box	Click on cancel button of confirmation box	Donor.java file not deleted	Donor.java file not deleted	Pass
VC_41	Check Back button	Click on Back button	Redirect to home page	Redirected to home page	Pass

PROGRESS REPORT PAGE

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
PR_1	SYNCHRONISED WITH WBS	TASK NAMES FILLED IN WBS	TASK NAMES FILLED IN WBS WILL BE SHOWN IN PROGRESS PAGE	TASK NAMES FILLED IN WBS WILL BE SHOWN IN PROGRESS PAGE	PASS
PR_2	INDIVIDUAL SUBTASK COMPLETION	EACH SUBTASK SUBMISSION REPORT	SHOWS WHETHER SUBTASK IS DONE OR NOT	SHOWS WHETHER SUBTASK IS DONE OR NOT	PASS
PR_3	TASK COMPLETED % BAR	NO. OF SUBTASKS DONE OUT OF TOTAL SUBTASKS	SHOWS % OF TASK DONE IN FORM OF BAR GRAPH	SHOWS % OF TASK DONE IN FORM OF BAR GRAPH	PASS
PR_4	CIRCULAR PROGRESS REPORT	TOTA PROJECT COMLPETED	SHOW TOTAL % OF PROJECT COMPLETED AS A WHOLE	SHOW TOTAL % OF PROJECT COMPLETED AS A WHOLE	PASS
NB	FUNCTIONAL NAVIGATION BAR	NAVIGATION BUTTONS	CLICKING BUTTONS WILL NAVIGATE US TO OTHER PAGES	CLICKING BUTTONS WILL NAVIGATE US TO OTHER PAGES	PASS

WBS

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
WBS_1	Click on add node button	Add a new node in the WBS	A new node gets add below the child node	A new node gets add below the child node	Pass
WBS_2	Add the value of new node in the dialog box	Task to be done (Eg: Finish module 1, etc)	The new node sets its value as message	The new node sets its value as message	Pass
WBS_3	Right click on any node to change its value	Edit the current task to new desired value	The new value replaces the old value in selected node	The new value replaces the old value in selected node	Pass
WBS_4	Click on delete node button	Delete the selected node from the WBS	The node which is selected should get deleted and removed from WBS	The node which is selected should get deleted and removed from WBS	Pass

7.Cost Analysis

Since this project is a web server hosted program, the only expenses are the cost of web hosting,managing databases which can be collected through advertisements on the website. This will be free of cost for the client/end-user. Moreover, revenue can be generated in the long term period.

8.RESULT AND DISCUSSION

Our project management tool tries to bring in all the necessary functionalities required when a group of people works on one project under one application. You can set tasks, assign them, track the progress and update the progress from one application, in order to track big commits and changes in the codebase the members can go around and get the history of all commits made in the project. The members can also get the username of the person who was responsible for authoring/pushing the commit. The main objective to make this project was to bring in all tools necessary for a developer to track changes in big team projects, we ourselves has faced problems in which tracking the changes was very difficult.

REFERENCES

https://reactjs.org/
https://nodejs.org/en/docs/
https://docs.mongodb.com/
https://mongoosejs.com/docs/api.html
https://docs.npmjs.com/